Journal of Computer Security 00 (2009) 1–38 1
DOI 10.3233/JCS-2009-0367
IOS Press

Exploiting cryptography for privacy-enhanced access control: A result of the PRIME Project

Claudio A. Ardagna [a], Jan Camenisch [b], Markulf Kohlweiss [c], Ronald Leenes [d], Gregory Neven [b], Bart Priem [d], Pierangela Samarati [a], Dieter Sommer [b] and Mario Verdicchio [b,*]

[a] *Università degli Studi di Milano, Milano, Italy*
[b] *IBM Zurich Research Laboratory, Zürich, Switzerland*
[c] *Katholieke Universiteit Leuven, Leuven, The Netherlands*
[d] *Universiteit van Tilburg, ???*

We conduct more and more of our daily interactions over electronic media. The EC-funded project PRIME (Privacy and Identity Management for Europe) envisions that individuals will be able to interact in this information society in a secure and safe way while retaining control of their privacy. The project had set out to prove that existing privacy-enhancing technologies allow for the construction of a user-controlled identity management system that comes surprisingly close to this vision. This paper describes two key elements of the PRIME identity management systems: anonymous credentials and policy languages that fully exploit the advanced functionality offered by anonymous credentials. These two key elements enable the users to carry out transactions, e.g., over the Internet, revealing only the strictly necessary personal information. Apart from presenting for the first time these two key results, this paper also motivates the need for privacy enhancing identity management, gives concrete requirements for such a system and then describes the key principles of the PRIME identity management solution.

Keywords: **???**

1. Introduction

Almost everyone uses electronic means for their daily interactions with businesses, governments, colleagues, friends, and family. In these interactions we play different roles such as customer, citizen, patient, and family member and we disclose personal information ranging from attributes such as date of birth, age, and home address to credentials pertaining to skills and rights. Indeed, the number of transactions we conduct electronically is ever growing and in fact not limited to those over the Internet as electronic authentication and authorization with some kind of token (e.g., electronic identity cards, driver's licenses, tickets and toll-tokens) become wide spread.

[*] Visiting researcher from Università degli Studi di Bergamo.

In our non-electronic lives, we naturally play different roles and display different faces of ourselves and typically only reveal partial information about ourselves. We give specific performances to specific audiences and try to keep these audiences segregated [38]. The capability to keep audiences apart and reveal different aspects of oneself in different contexts is an essential characteristic of our lives [57]: "[T]he sort of relationship people have with one another involves a conception of how it is appropriate for them to behave with each other, and what is more, a conception of the kind and degree of knowledge concerning one another which it is appropriate to have". (p. 328) Social relationships require a certain amount of privacy or, as poet Frost wrote, "Good fences make good neighbors".

This role playing and presentation of self is part of our identity. Identity in this light is not some innate quality, but the result of publicly validated performances, the sum of all roles played by the individual [38]. Individuals have a number of partial identities that allow them to name and sort themselves, to adjust themselves to social contexts, to have a plural social life, to be part of the public realm, and to align their own perceptions on identity with the perceptions of others [36,56].

Of course, information occasionally crosses the borders of social contexts, usually much to the chagrin of the individual involved, but by and large most people master handling their partial identities in the offline world. Now, we are challenged to apply the same skills for electronic transactions because digital data is much easier to store and process and is hardly ever deleted or forgotten.

Clarke's [27] notion of the digital persona as "a model of an individual's public personality based on data and maintained by transactions, and intended for use as a proxy for the individual" is helpful to understand why managing one's identity and controlling ones personal data has become a crucial ability in the online world as well. The individual has some degree of control over a *projected persona*, the image the individual wants to portray of herself, but it is harder to influence the *imposed personae* that are created by others. Individuals maintain multiple projected digital personae, much like the different roles that they play in their offline life (partial identities). There are also multiple imposed personae that relate to a particular individual, because there are multiple entities who each create and maintain their own imposed personae. Projected and imposed personae, whether true or false, are used to make decisions regarding the interaction and treatment of the individual (see [41,76,77]). Different digital personae are also combined into richer composite digital personae which replace the original partial digital personae. This easily leads to the undermining of audience segregation and decontextualization of information. The context in which an individual reveals certain aspects of their identity matters. Simply combining contextual data into a super persona neglects the essential characteristics of identities. For instance, in one context, one might be a good (nice) guy (teaching), while in another, one may (professionally) be a bad guy (judging). Both aspects are part of this individual and cannot be averaged.

The individual therefore needs to be able to manage their online identities, just like in the offline world. The technical complexity, the volatile nature of the media, and its

rapid changes make this a non trivial task and therefore the support of technological means such as identity management systems is essential.

Traditionally, online identity management (IdM) is driven by organisations for purposes of controlling access to resources. This IdM perspective focuses on the strategic objectives of the enterprise aiming at reducing the risks of data loss, ensuring the accuracy of identity information, utilizing the storage and management of personal data, and using information for the efficient development and distribution of products and services [49]. Similar needs hold for government-driven IdM systems, where, for example, delivering efficient electronic public services without the risks of fraud and insecurity are central goals.

Organisations provide resources and endeavor to control access to these resources. Access control – i.e., identification, authentication and authorisation – thus is one of the key functions of identity management from the perspective of enterprises, although it is interweaved with other functions (see Fig. 1). Only properly authorised entities (e.g., clients, known customers) are allowed to make use of the requested services. Identity management systems in this context maintain digital identities, or accounts, containing attributes (e.g., a name) and properties (e.g., entitlements within the system's domain such as access rights) of the entities (usually individuals) within their domain. The accounts have an identifier (e.g., username) and one or more authenticators (e.g., a password).

The individual's need for control over the presentation of self and audience segregation are neglected in these traditional systems.

More recently a shift in focus of identity management can be witnessed from a strict perspective of enterprise-centric access control to resources towards a perspective that takes the interests of the individual into account. A number of iden-

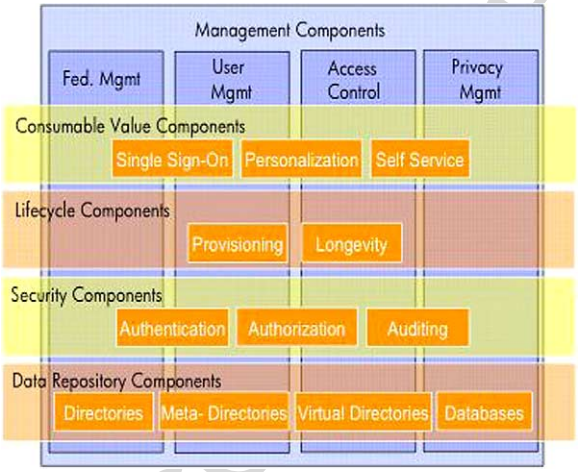

Fig. 1. Current IdM solution stack (taken from [54], illustration by Jan De Clercq (HP) and Marco Casassa Mont (HP Labs)).

tity management systems are available today, being standardized, or developed that are illustrative for this change in focus. These include the open source project Higgins, Microsoft's CardSpace, the web services standards, and the Liberty Alliance set of protocols. These initiatives provide individuals support in managing their online identities. They primarily focus on identity management and less on privacy preservation and identity management as outlined above.

The PRIME project has showed that privacy has to be taken seriously in online identity management and that identity management indeed can be done in a way that provides maximal privacy to the users, applying the state of the art privacy enhancing technologies. This paper illustrates this by presenting two probably most important technical components of the PRIME project: anonymous credentials with various extensions as required for many practical scenarios and a policy language that uses these concepts and thus enables system designers to take advantage of the cryptographic mechanisms to protect the users' privacy.

The remainder of this paper is organized as follows. In the next section, we briefly discuss the core principles and requirements for privacy enhancing identity management as they are defined in the PRIME project. Next we provide a use case of privacy enhanced identity management and summarize the main technical components of the PRIME identity management solution. The remainder of the paper discusses anonymous credentials with their various extensions and privacy-enhancing policy languages. For the anonymous credential system we do not provide the cryptographic details on how to realize them but rather present them at an abstract functional interfaces suitable for the policy language. The conclude with related work and an outlook.

2. Privacy-enhancing identity management

Incorporating privacy enhancing functions into IdM systems is difficult because privacy and identity are complex concepts and one has to balance the various interests in a delicate way; privacy-enhanced IdM systems still need to facilitate online interaction in a way that satisfies both enterprises and individuals. As a starting point for development the PRIME project has established a number of design principles to meet this challenge. The project has moreover elaborated these principles into concrete requirements. We describe both in the following.

2.1. Principles

The primary design principle states that IdM systems need to start with maximum privacy, so users can make autonomous choices about the use and construction of identities from an anonymous realm. Secondly, IdM systems need to be governed by specific privacy policies that must not only be stated, but also be enforceable by technical means. Of course, enforcement needs to be trustworthy, which means

that the computing platform on which the IdM technology is being built needs to be trustworthy, and that external trust mechanisms should assure compliance with law and policies. IdM systems furthermore need to be useable by non-expert users, and thus need to provide easy and intuitive abstractions of privacy. The models employed by the technology need to be hidden for the user. Finally, PRIME acknowledges that privacy-enhanced solutions need to be integrated into new applications [54].

Next to the PRIME principles, existing legal principles considering the processing of personal data have also provided guidance for the development of PRIME solutions. PRIME solutions are designed to comply with the current EU legal framework.[1] These legal data protection principles predominantly state that processing of personal data needs to be fair and lawful, and that data may only be collected as far as it is necessary to meet a specified and legitimate purpose. Moreover, the personal data collected must be restricted to the minimum sufficient for this purpose, and data must not be kept longer than necessary (data parsimony). In addition, the provisions in the Directive state that prior to the processing of personal data, the national data protection authority needs to be notified and that data may only be transferred to non-European Union countries if these ensure an adequate level of protection [55].

2.2. Concrete requirements for user privacy

The preceding principles have been elaborated and extended into more detailed, user-centric requirements for privacy-enhanced IdM systems, which can be divided in requirements pertaining to 'audience segregation through user control', and requirements for 'user adoption'.

Audience segregation can be achieved by facilitating the construction and deployment of different partial identities under control of the user. User control implements a core aspect of informational privacy (see [35,57,68]). Within PRIME, user control is decomposed into five sub-requirements: *information*, *consent*, *access*, *correction*, and *security*. They capture a number of legal and social requirements [54,55] and can also be found in other guidelines for privacy-enhancing IdM systems (e.g., [44] and [52].

In exercising control a prerequisite is to have *information* relating to aspects such as data controller and data collection purpose. This information enables the individual to make well-considered decisions about the use of identities and data to be disclosed. This requirement translates into an obligation for organisations to use IdM systems that communicate these aspects in an understandable form. Providing proper information relating to data collection and use helps to improve the predictability and consistency of an IdM system and the services it facilitates. Properly informed users can make informed choices which, in the absence of undue influences, translates into *informed consent* to processing of certain personal data. Consent thus must be voluntary and, ideally, revocable. It needs to relate to a specific use of personal data,

[1] See: Art. 6(1) Dir. 95/46/EC, Art. 18 Dir. 95/46/EC and Art.25 Dir 95/46/EC.

and should be given explicitly when sensitive data is processed. Consent implies that so-called 'take-it-or-leave-us' approaches are undesirable; users should have real choices concerning their identity and the data they disclose in a particular interaction. Moreover, they need to be able to define the boundaries of data use, for instance by stating and assigning policies to certain data. This aspect of 'confinement' is necessary to avoid extensive use of personal data.

After personal data is disclosed to a service, users need to be able to inspect (*access*) their data, because user control would be a useless concept when data held by the server can not be inspected for errors and abuse. Thus, actions of data collectors need to be transparent, for instance through notification of data processing. This limits power imbalances between individual and data collectors. Transparency of data processing should concern the whole chain of organisations that use data regarding a service.

Individuals should be able to have their personal data corrected or erased to some extent, for instance when mistakes are made or decisions are regretted. The online world, after all, does not provide the level of 'forgetfulness' customary in the offline world [11]. Because the lives of people change, mechanisms to realign digital identities to real life identities are necessary. IdM systems need to facilitate this and should therefore provide features for correction, objection, and erasure of personal data. *Security* is also a condition for user control because control will certainly be lost when personal data inadvertently leaves the realm of the data controller. IdM systems need to have appropriate security measures, which need to be displayed to the (non-expert) user by means of understandable and appropriate trust markers.

Trust also relates to another important requirement for privacy-enhanced IdM, which is 'user adoption'. The feasibility of Privacy-enhancing IdM depends on a critical mass of users. Privacy-Enhancing-Technologies (PETs) are not yet widely adopted, and the readiness of people to invest in PETs seems low [31,63]. To increase adoption, privacy-enhancing IdM systems therefore must be flexible in the sense that they can be used by everyone (to limit risks of creating 'digital divides' in the field of privacy-protection), should be adaptable to social settings, and have a reasonable price. The price people are willing to pay and the efforts they are willing to make for privacy-enhancement, depends on the sense of urgency and the general conception about privacy risks on the Internet. Because of this, the added value of a privacy-enhancing IdM system needs to be understandable to the user [62].

3. A scenario of PRIME-enabled online shopping

PRIME has built an identity management solution that realizes the above requirements. It basically consists of a set of components that all the involved parties use to conduct their transactions. Before we describe the solution in the next section, we illustrate the principles and requirements discussed in the previous section with a

simple online shopping scenario, featuring Alice, and at the same time show how the PRIME solution is applied.

After a recommendation from her sister Alicia, Alice considers to purchase a box of white wine at 'CyberWinery.com'. Figure 2 shows the main entities involved in the scenario and the data flows in the non PRIME-enabled situation. For example, prior to the purchase Alice is likely to first create an account at CyberWinery, thereby disclosing personal data. The account will store purchase data, personal preferences, and possibly even credit card data. CyberWinery has outsourced warehousing and delivery to 'LogisticsProvider', which requires data from CyberWinery (like a delivery address). CyberWinery will request 'CreditProcessor' to authorize Alice's credit card which leaves traces at 'CreditProcessor' because they will store the transaction details for their own business and accounting purposes. Other services may also be present, such as CyberBooks which is recommended by Alicia for the purchase of the Good Wine Guide. This purchase again requires Alice to register with her personal data and possibly CreditProcessor and LogisticsProcessor are also involved in this transaction.

Alice, a vigilant and sensitive 'Netizen', is aware of the risks involved in online transactions and knows that the loss of personal information can cause severe financial and reputational damages which are difficult to repair, and she has heard that the use of personal data by others may lead to discrimination, exclusion, and social sort-

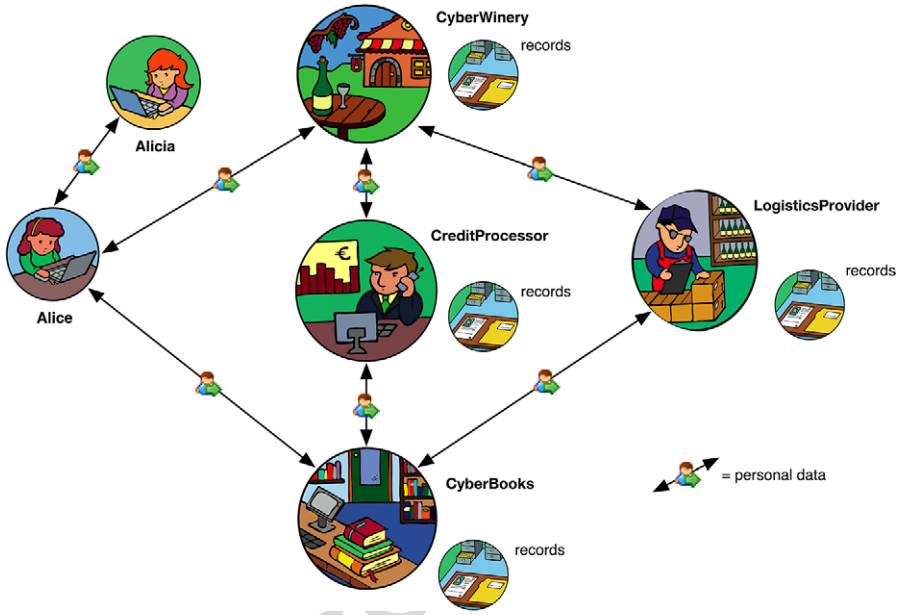

Fig. 2. Traditional user data exchange in an online shopping scenario (taken from [46], illustration by Tjeerd van der Hulst).

ing. Because of this, she adheres to the principle to share a minimum amount of personal data on the Internet. Fortunately for Alice, CyberWinery is a PRIME-enabled website, which assures her that she can make use of a secure privacy-enhancing identity infrastructure that complies with current data protection legislation. CyberWinery's PRIME-enabled website has several features that ensure privacy and security throughout the whole shopping process. For example, Alice trusts CyberWinery because it uses PRIME technology. She has read about PRIME and has completed the PRIME online tutorial.[2] CyberWinery also has respected trust marks and provides clear information about the buying process. In line with the PRIME principles, the shop displays its physical location and states the purposes of data collection in simple non-technical privacy policies. Alice can inspect the more detailed and technical privacy policies if she wants to.

Alice proceeds with her purchase. Unlike many other web stores, CyberWinery lets Alice determine her own requirements for the data exchange. Instead of providing a 'take it or leave us' privacy policy, CyberWinery allows Alice to control her online identity. Her wish to share only a minimum amount of data is facilitated by the possibility to do anonymous or pseudonymous purchases. This is even a default setting. CyberWinery still demands Alice to prove that she is creditworthy and over 18, but this is possible even within Alice's choice to be pseudonymous. Alice only needs to attribute a number of private PRIME-credentials (issued by Trusted Third Parties, such as her bank or the State) to her chosen pseudonym.

On entering the CyberWinery website, Alice's PRIME-console (implemented as a browser extension and middleware) takes over the negotiation process concerning the use of personal data. The Console helps Alice make informed decisions and takes over certain tasks on the basis of her prior preferences. After negotiating data handling policies, Alice consents to the disclosure of certain data compliant with her policies: information may not be used for unsolicited communication or shared with business affiliates. The Console makes it possible not only to state these preferences, but also associates these requirements to the data (by means of 'sticky policies'). Thus, Alice can easily approve and confine the use of her data by CyberWinery and have her policies enforced at their end. Alice also uses the PRIME Console to create an encrypted token containing her address that can only be decrypted by LogisticsProcessor; there is no need for CyberWinery to know the delivery address.

Alice has ordered a box of white wine and wants to check the delivery date and the data stored about her at CyberWinery. The PRIME Console on her computer and the PRIME Middleware at CyberWinery provide this option. The Console allows her to keep track of the information CyberWinery has about her, without her having to remember the identity she used for the transaction. The Console provides a comprehensive history of all her online transactions (involving her many online identities) and she can check the enforcement of her policies and she can intervene when she

[2] See: www.prime-project.eu/tutorials.

detects errors and abuse. While checking the delivery date, she notices that LogisticsProvider has requested her encrypted delivery address, and that this address has automatically been deleted by the PRIME Middleware, according to her negotiated policy.

After a few days, LogisticsProvider delivered the wine. Alice, pleased with its quality, becomes a returning customer and decides to host a wine party for her friends. Alice, being only an expert in white wines, decides to sign up for CyberWinery's recommendation tool for wines that meet her taste and budget. The PRIME Console helps her create pseudonyms that guarantee unlinkability between the pseudonyms used for her prior purchases and that used for the recommendation system while maintaining her wine preferences. This allows Alice to keep control over the profiles that CyberWinery creates for her pseudonym. At the same time, CyberWinery can benefit from the provision of a personalised and tailor-made service to Alice when she allows this.

Cyberwinery enables Alice to use different pseudonyms in a convenient way. This makes it possible for her to avoid linkability of her purchases at CyberWinery with other activities, like her activities on a wine-forum, or the purchase of a wine guide. CyberWinery allows the segregation of contexts Alice engages in. With PRIME Middleware, Alice can even seamlessly change contexts inside one single session, using for example different pseudonyms in her roles as 'buyer' and 'browser'. Alice's use of pseudonyms is guaranteed by PRIME technology, which communicates her data using public key encryption. Hence, eavesdroppers cannot intercept any of her private information, and false websites can be detected by PRIME Middleware. The store does not use cookies to unobtrusively link interactions that Alice wants to keep separated.

After a while, Alice becomes a white wine expert and she likes to discuss wines on a weblog she visits frequently, called iConnoisseur, where she gained a reputation. Normally, it would be difficult to transfer such a reputation to other online contexts, without underlying activities becoming linkable. PRIME technologies provide an answer to this issue by the possibility to create and issue credentials. PRIME-enabled weblogs like iConnoisseur can create 'reputation-credentials' that cannot be tampered with and can issue a reputation to Alice, which she can subsequently use in other contexts, like the website of CyberWinery.

The PRIME-Middleware credentials can also assure the accountability of people and organisations. Anonymity and pseudonymity have their limits and even though Alice is reliable, there may always be people that want to abuse a privacy-enhanced website. When CyberWinery detects fraud or contractual default it can ask the credential provider used to create the anonymous credentials used by a particular customer to revoke their anonymity.

4. The PRIME solution

In this section, we first describe the technical principles that PRIME employs to realize a privacy-enhancing user-centric identity management system, i.e., to help

Alice protecting her privacy. We then continue with a short description of the PRIME architecture embedding all these technical building blocks and finally describe the interaction between an Alice and the Wineshop to illustrate how the PRIME system works.

4.1. Privacy-enhancing technologies for identity management

The principle of *data parsimony* is the driving principle of our system: *no party should per se learn any information other than what it absolutely needs* to conduct a transaction or, more generally, for the purpose at hand. Determining which information this is depends of course on the particular application and on the business and legal requirements. However, such decisions often also depend on what particular technology is used to implement the application or business process. Therefore, PRIME has built a system that brings together the state-of-the-art privacy-enhancing technologies that indeed allow one to implement the principle of data parsimony and does not require users to provide additional identity information only because of the imperfection of the technology. The PRIME solution employs the following mechanisms to achieve this.

Anonymous communication: First of all, the communication layer needs to be secure and anonymous (i.e., it must not reveal potentially identifiable information such as the user's IP address or location). This can be met by so-called mix networks or onion-routing systems, e.g., [9,33].

Private credentials: Whenever the user needs to provide some (certified) information about herself, private credential systems [14,17,22] allow her to use her certificates to selectively reveal certified attributes about herself. For instance, if the user is to reveal that she is a teenager, she should not be required to provide her name or even her exact birth date! Moreover, the party who certifies that a user is of age and the party who verifies the statement should not be able to tell whether they communicated with the same user or with different ones.

Attribute-based access control: Access to resources or service is given on a basis of properties or attributes of the user. Thus, per resource one needs to specify which attribute a user needs to have in order to access some information. Also, this specification needs to be done such that the user is required only to reveal the information about herself that is necessary to decide whether or not she is entitled.

Data handling policies: When a user request access to some resource or service, she is informed about the access control requirements, i.e., what information she needs to provide, but also the data handling guarantees, i.e., how her data will be handled. Once the user has revealed her data, then this *data handling policy* is stored together with the data, and is then enforced by the service provider (which includes handling obligations such as deleting the data after a certain amount of time).

User interfaces: The PRIME solution includes a user interface that lets the user manage her different identities, to see what data is released under what conditions and to whom (so the user can give informed consent), and to view past transactions.

Realizing privacy-enhancing identity management in practice not only requires that the technologies listed above are employed but in many cases also requires that third-party services are available, including privacy-enabling infrastructure services such as identity brokers, traffic anonymizers (e.g., running nodes of TOR or JAP), and all kinds of certification authorities.

4.2. The PRIME middleware

The PRIME system [53] is basically a middleware architecture consisting of different components implementing the mechanisms described above. All parties conduct their transactions through the middleware. This is necessary because at the users' end, the PRIME middleware needs to control all releases of the users' data. All the users' data (including identity information and credentials) are therefore stored in a database that is protected by the PRIME middleware. At the service providers' side, the situation is similar as the data could potentially be personal data of users which needs to be protected by access control and data handling policies. Thus, the PRIME architecture is symmetric and all involved parties apply essentially the same components.

The PRIME architecture can be seen as a blueprint that defines a possible way of bringing different technologies from the PET space together with the goal of improving the privacy protection for people that interact over an electronic communication network such as the Internet.

The PRIME architecture can be seen as a blueprint that defines a possible way of bringing different technologies from the PET space together with the goal of improving the privacy protection for people that interact over an electronic communication network such as the Internet. The PRIME architecture integrates mechanisms that cover the protection of privacy throughout the life cycle of personal data. The core of the PRIME architecture is its machinery for performing identity federation in a privacy-enhanced way, the most important components thereof being the policy languages and cryptography described in this paper. The PRIME architecture also addresses the part of the data life cycle after a party has authenticated itself, that is, has revealed identity attributes to another party. For addressing this part of the data life cycle, we feature an architectural component for privacy obligation management. It is driven by policies that have been agreed on with the parties having released the data.

4.3. PRIME at work

We now describe the protocol flows between Alice and the Wineshop and how this involves the privacy-enhancing mechanisms, in particular the ones described in the following sections.

We start with Alice ordering her box of wine: Alice's request for the wine triggers the webstore's access control component. This component checks whether Alice is allowed to access the resource (the box of wine) and, as Alice has not yet sent any information about herself in this transaction, the component responds by sending a request for a claim satisfying the condition in the access control policy (ACP) for the requested resource. In this example, the ACP could be that the customer needs to show that she is over 18 years of age. She is offered the choice to provide proof by means of a valid OECD ID document and an encrypted copy of her name and address as appearing on the OECD-approved ID document. Alternatively she could present a pseudonym established in a previous transaction. The ACP also contains statements about how the data revealed by Alice will be handled by the receiving party (i.e., the Data Handling Policy).

Thus, Alice's PRIME middleware access control component will receive the claim request, i.e., the ACP. In response, the component will make a release decision whether (and possibly which of) the requested claims will be provided to the service provider and under what conditions. To his end, it will evaluate whether Alice possesses the necessary (private) credentials to satisfy the request. For this to work out, the OECD ID passport may be instantiated with a Swiss passport, and the address on an OECD Photo ID may be instantiated with the address as appearing on Alice's Swiss driver's license. Ontologies are used to ensure that these instantiations are correct.

If the service provider is unknown to Alice's PRIME middleware, it may first issue a request to the shop to prove that it meets certain requirements such as complying to certain standards (e.g., whether the shop possesses a privacy seal such as TRUSTe). This (optional) request is similar to the shop's request for proof of Alice's legal age. If the shop provides such a proof, it will be verified and logged. If Alice's access control component then decides that the requested claim can be released, Alice is presented via the PRIME user interface with a selection of credentials that she can use to satisfy the request, a summary of what data the service provider requests and for what purpose. If Alice decides to go on with the transaction, the claim and evidence will be communicated to the service provider. The claim is the ACP, potentially modified by Alice, and the evidence consists of all kinds of credentials and certificates that back the claims.

The modified claim may for instance state that the encrypted name and address may be provided to the shipping service for the purpose of being able to ship the order and the data may be retained for a maximum of three years or whatever is legally obligatory.

Alice's PRIME middleware will next log which data has been disclosed under which conditions. This enables Alice to view her transaction records and, with support from her system, to judge to extend to which the released data allow one to identify her.

After receiving the data requested from Alice, the service provider verifies the claims and if this succeeds, grants Alice the resource requested. The service provider further stores Alice's data together with the agreed policies to that they can be enforced.

5. Anonymous credentials for privacy-enhanced policy languages

We now describe the first technical key ingredient of privacy-enhanced identity management, i.e., anonymous credentials and their various extensions. While the basic concept has been known for quite some time [14,17,22], efficient realizations are still quite new and only recently many extensions important to their practical applications have been invented. Many of the extensions are results of the PRIME project. In this section we give for the first time comprehensive descriptions of these advanced features and unify them into a single system. We do so without going into mathematical details; rather, we give a simplified and unified view of the high-level application interface that is offered by the various cryptographic building blocks. We also introduce more complex elements to define a privacy-enhanced policy language.

5.1. Concept of anonymous credentials

Anonymous credentials can be thought of as digitally signed lists of attribute-value pairs that allow the owner of such a credential to prove statements about attribute values without revealing any more information about them than what is directly implied by the statement.

Classical *digital signatures* [40,58] allow a signer to authenticate digital messages using a secret key *sk* that only she knows. The corresponding public key *pk* is made known to the world, for example by publishing it in a public directory, so that anyone can verify the validity of signatures issued using *sk*. Digital signatures are *unforgeable*, in the sense that no adversary can create a valid signature on a new message, even after having seen a number of valid signatures on other messages.

A credential is essentially a digital signature issued by a trusted authority on an ordered list of attribute-value pairs $(A_1 = a_1, \ldots, A_n = a_n)$.[3] By issuing a credential, the authority certifies that the user satisfies the described attributes. For example, the

[3] It is also possible to have credentials signed by the user himself (*pk* is the user's public key) or not signed at all (*pk* is the empty string, called *declarations* in [12]).

government authorities could issue electronic identity cards in the form of credentials under the government's public key pk_{G} on a list of attribute-value pairs

$$(\texttt{name} = \text{“}\texttt{Alice}\text{”}, \texttt{bdate} = 1968/05/27,$$
$$\texttt{address} = \text{“}\texttt{15 A Street}, \texttt{Sometown}\text{”}).$$

The most obvious way for a user to convince a verifier that she owns a valid credential for a certain set of attributes would be to simply send the credential (i.e., the list of attribute values and the signature) to the verifier. A slightly more advanced approach would be to include the user's public key as an attribute in the credential, so that the user can authenticate himself as having the correct attributes using the corresponding secret key. Both approaches have the major disadvantage however that the owner of the credential has to disclose *all* attributes in the credential in order to authenticate himself, since otherwise the authority's signature cannot be verified.

Anonymous credentials provide a privacy-friendly alternative. Namely, they allow the user and the verifier to engage in an interactive *selective-show* protocol during which the user proves that she owns a valid credential of which the attribute values satisfy some *statement*. The only information leaked about the attribute values however is that the statement holds true. For example, if Alice uses the credential above to prove the statement $\texttt{address} = \text{“}\texttt{15 A Street}, \texttt{Sometown}\text{”}$, then her name and birth date remain hidden from the verifier. If she proves the statement $\texttt{bdate} < 1990/01/01$, then her name, her address, and even her exact date of birth remain hidden: all she reveals is the mere fact that it is before 1990.

5.2. A language for anonymous credentials

In the past, credentials have been exploited to take decision on whether a given party may or may not access a service. Today, anonymous credentials represent an important driver towards the definition of a privacy-enhanced policy language. In Fig. 3, we introduce a grammar of a language that allows to describe complex expressions over (anonymous) credential attributes. In the remainder, we illustrate the grammar elements that refer to anonymous credentials in more detail.

5.2.1. Selective showing of attributes

Anonymous credentials come with a *selective-show protocol*, which is an interaction between the user and a verifier, during which the user cryptographically proves to the verifier that she owns a set of credentials satisfying some claim over the attributes. The security of the protocol guarantees that a cheating user cannot successfully convince a verifier of a false claim, and that the verifier learns nothing more about the attribute values than what is implied by the claim.

If a user has credentials $cred_1, \ldots, cred_\ell$, where $cred_i$ authenticates attribute-value pairs $(A_{i,j} = a_{i,j})_{1 \leqslant j \leqslant n_i}$ issued under pk_i, $1 \leqslant i \leqslant \ell$, then the input–output behavior of the cryptographic selective-show protocol is given by:

$$
\begin{aligned}
\langle exp \rangle &::= \texttt{cred\_type}^{pk}[A] \mid s \mid n \mid n \cdot \langle exp \rangle \mid \langle exp \rangle + \langle exp \rangle \\
\langle math \rangle &::= < \mid \leqslant \mid = \mid \neq \mid \geqslant \mid > \mid \in \\
\langle cond \rangle &::= A \mid A \langle math \rangle a \mid \mathit{NymDer}(nym, A) \mid \mathit{SerialDer}(S, A, context, limit) \\
&\quad \mid \mathit{EscrowShare}(ess, S, A, context, limit, \langle exp \rangle) \\
\langle condlist \rangle &::= \langle cond \rangle \mid \langle cond \rangle, \langle condlist \rangle \\
\langle logic \rangle &::= \wedge \mid \vee \\
\langle claim \rangle &::= \texttt{cred\_type}^{pk}[\langle condlist \rangle] \mid \langle exp \rangle \langle math \rangle \langle exp \rangle \\
&\quad \mid \mathit{VerEnc}(C, pk, \langle exp \rangle, \lambda) \mid \langle claim \rangle \langle logic \rangle \langle claim \rangle
\end{aligned}
$$

Fig. 3. Backus–Naur form of complex expressions over attributes.

Selective-show protocol:

Common input: $pk_1, \ldots, pk_\ell$, $claim(A_{i,j})$

User input: $cred_1^{pk_1}, \ldots, cred_\ell^{pk_\ell}$ — Verifier input: none

User output: none — Verifier output: accept/reject

In theory, efficient protocols exist for all computable claims using generic zero-knowledge techniques [39]. However, the protocols thus obtained are usually too expensive for practical use. We therefore restrict the expressivity of the claims to operations for which truly efficient protocols exist. Below we give an exhaustive list of such operations.

Reveal: $A_{i,j} = a_{i,j}$. This is the simple operation where the user discloses to the verifier the value $a_{i,j}$ of an attribute $A_{i,j}$.

Equality: $A_{i,j} = A_{i,j}$. The user shows that two attributes, possibly from different credentials, are equal – without disclosing their value.

Comparative: $A_{i,j} < c$, $A_{i,j} > c$. Prove that an attribute is less or greater than a constant c. In fact, one can also prove inequality of two attributes, and use any of the operators $<, \leqslant, =, \geqslant, >$.

Interval: $exp \in [c_1, c_2]$. Prove that an expression of attribute values is within a given interval.

Simple arithmetic: $A_{i,j} + c$, $A_{i,j} \cdot c$, $c_1 \cdot A_{i,j} + c_2 \cdot A_{i',j'}$. Not just attributes and constants can be compared, but also simple arithmetic expressions over attributes (essentially, sums of products of an attribute with a constant).

Logical: $claim_1 \wedge claim_2$, $claim_1 \vee claim_2$. Prove that the logical conjunction or disjunction of two claims is true. Again, no other information is leaked to the verifier. In particular, in case of a disjunction, the verifier does not learn which of the two claims is true.

The functionality of the selective-show protocol is captured in the policy language through the expressions and the claims dealing with credential attributes, as defined below.

Definition 5.1 (Credential attribute expression). If *cred* is a credential of type `cred_type`, signed under the public encryption key *pk*, comprised of attributes $A_1, \ldots, A_n$, then $\texttt{cred\_type}^{pk}[A_i]$ (with $1 \leqslant i \leqslant n$) is an expression that refers to attribute A_i in *cred*.

Let $Math$ be a set of symbols representing standard mathematical predicates (e.g., '$=$', '$\neq$', '$>$'). We introduce *credential claims* to express restrictions on the values of the attributes in a credential, as follows.

Definition 5.2 (Credential claim). Given a public key *pk*, an attribute A, and a value a, a credential claim $\texttt{cred\_type}^{pk}[A \; math \; a]$, where $math \in Math$, refers to a credential of type `cred_type`, signed under *pk*, of which attribute A satisfies the restriction expressed by $A \; math \; a$.

Credential claims will be used in the policies to express the need for the user to demonstrate that she possesses credentials satisfying the required restrictions. To illustrate the above definitions, we now give a number of examples of claims expressed in our policy language.

Example 5.1. The claim $\texttt{identity\_card}^{pk_{\mathrm{G}}}[\texttt{name} = \text{“}\texttt{Ross}\text{”}, \texttt{bdate} < 1991/01/01]$ denotes a credential of type `identity_card` signed under the government's public key pk_{G} whose attribute `name` has value "`Ross`" and its `bdate` attribute is a date before 1991, meaning that the subject is over eighteen.

Example 5.2. In the wine shop example, suppose that Alice, in addition to her identity card credential that we mentioned above, also has a credential under her bank's public key pk_{B} authenticating her credit card information with

$$(\texttt{name} = \texttt{Alice}, \texttt{bdate} = 1968/05/27, \texttt{cardnr} = 123456, \\ \texttt{exp} = 2012/07, \texttt{pin} = 1234).$$

Alice may not want to reveal her identity when purchasing a box of wine, but the shop may require her to reveal her address and credit card information, and to show that she is over 18 years of age and that the credit card is registered on her own name – without revealing her name. She does so by engaging in a selective-show protocol with the wine shop showing the claim

$$\texttt{identity\_card}^{pk_{\mathrm{G}}}[\texttt{bdate} < 1991/01/01, \\ \texttt{address} = \text{“}\texttt{15 A Street, Sometown}\text{”}] \\ \wedge \texttt{credit\_card}^{pk_{\mathrm{B}}}[\texttt{cardnr} = 123456, \texttt{exp} = 2012/07] \\ \wedge \texttt{credit\_card}^{pk_{\mathrm{B}}}[\texttt{name}] = \texttt{identity\_card}^{pk_{\mathrm{G}}}[\texttt{name}]$$

5.2.2. Pseudonymous identification

When a user regularly accesses the same service, she may not want to reprove at each visit that she qualifies for using the service, but may prefer to establish a permanent user account instead. To protect her privacy, she wants to do so under a pseudonym: it is bad enough that all her actions at this service now become linkable, she does not want them to become linkable to her actions across other services too. The server, on the other hand, may want to prevent users sharing their account information with others, thereby giving non-qualified users access to the service as well.

The cryptography can help out here. A pseudonymous identification scheme allows a user to derive from a single master secret multiple *cryptographic pseudonyms*, and later authenticate herself by proving that she knows the master secret underlying a cryptographic pseudonym. The user first chooses a random master secret key *msk*. From this master secret, she can derive as many unlinkable pseudonyms *nym* as she wants. Later, using her master secret key *msk*, she can authenticate herself with respect to *nym*. The central idea is that *all* the user's credentials are underlain by the *same* master secret *msk*, so that by sharing *msk* with others, the user is sharing her *whole* identity, rather than just her pseudonym *nym* and the associated access to this service.

The underlying cryptography gives a double security guarantee. On the one hand, it guards against impersonation attacks, meaning that no user can successfully authenticate herself without knowing the underlying master secret. On the other hand, it guarantees that different pseudonyms are unlinkable, meaning that nobody can tell whether two pseudonyms were derived from the same master secret or not.

Of particular interest to identity management systems are those pseudonymous identification schemes that are compatible with an anonymous credential scheme, allowing the master secret key *msk* to be encoded as an attribute in the credential, and allowing the user to prove credential terms of the following form.

Definition 5.3 (Derived pseudonym predicate). The predicate $\mathit{NymDer}(\mathit{nym}, A)$ is true if and only if A encodes the master secret key from which the cryptographic pseudonym *nym* was derived.

Some anonymous credential scheme in the literature in fact realize pseudonymous identification (e.g., [22]).

Example 5.3. Alice's electronic identity card could have her master secret embedded as an attribute, so that her digital identity card is a credential under pk_{G} containing attribute-value pairs

$$(\texttt{name} = \text{“}\texttt{Alice}\text{”}, \texttt{bdate} = 1968/05/27,$$
$$\texttt{address} = \text{“}\texttt{15 A Street}, \texttt{Sometown}\text{”}, \texttt{msk} = \ldots).$$

When logging into the wine shop for the first time, she derives from the value in msk a fresh cryptographic pseudonym, sends it to the wine shop, and proves the claim

$$\texttt{identity\_card}^{pk_G}[\texttt{bdate} < 1991/01/01, \mathit{NymDer}(\mathit{nym}, \texttt{msk})]$$

to show that she has the proper age to buy wine. From that point on, she can log in under *nym*, so that the wine shop will recognize her as a registered customer with the required age.

5.2.3. *Verifiable encryption*

A public-key encryption scheme allows a sender to encrypt a plaintext message *m* under the public key of a receiver, so that only the receiver can decrypt the resulting ciphertext using the corresponding secret key. A *verifiable encryption scheme* [25] is a public-key encryption scheme that is 'compatible' with an anonymous credential scheme such that it allows claims to be proved about how the encrypted content was derived from attributes in a credential – without revealing the content.

The encryption algorithm additionally takes an extra input parameter called a *decryption label* λ. A ciphertext is not supposed to hide the label λ, but rather inseparably ties the label to the ciphertext so that the same label has to be used at decryption time, otherwise the ciphertext is considered invalid.

Verifiable encryption is used in identity management systems to provide a verifier with a ciphertext containing sensitive data about the user (e.g., her identity) under the public key of a trusted third party. In case of conflict or abuse, the verifier asks the trusted third party to decrypt the ciphertext. The label is used to describe the conditions under which the third party is allowed to decrypt the ciphertext. Since the label is inseparably attached to the ciphertext, these conditions cannot be changed or removed by a cheating verifier.

We extend our policy language with a predicate dealing with verifiable encryptions [8].

Definition 5.4 (Verifiable encryption predicate). Predicate $\mathit{VerEnc}(C, pk, \langle exp \rangle, \lambda)$ is true if and only if C is a ciphertext encrypted under public encryption key *pk* with decryption label λ carrying the value of the expression $\langle exp \rangle$ of credential attributes.

Example 5.4. In the wine shop example, Alice could use verifiable encryption to encrypt her address under the public key of a shipping company pk_S. She thereby hides her address from the wine merchant, but can still prove that what she encrypted is her real address. In the show protocol she proves the claim $\mathit{VerEnc}(C, pk_S, \texttt{identity\_card}^{pk_G}[\texttt{address}], \text{“}\texttt{shipping}\text{”})$ with respect to her identity card. The wine shop can then forward the ciphertext C to the shipping company, who can decrypt it and ship the box of wine to Alice.

Example 5.5. To speed up delivery, the wine shop already ships the wine before even the credit card transaction has been approved by the bank. In case something goes wrong, however, the wine shop wants to be able to revoke Alice's anonymity so that it can try to obtain its money in some other way. The wine shop therefore requires Alice to encrypt her name, as stated on her identity card, under the public key of a trusted third party (TTP) pk_{TTP}. This is specified in the policy using a predicate *VerEnc*(C, pk_{TTP}, `identity_card`$^{pk_{\text{G}}}$[`name`], "`failedpayment`"). In case of problems with the transaction, the wine shop contacts the TTP to have the ciphertext C decrypted, revealing Alice's identity.

5.2.4. Limited spending

Certain applications, such as e-cash or e-coupons, require that the number of times that a credential can be shown anonymously be limited. For instance, a credential representing a wallet of n coins can be shown n times. A user can nevertheless attempt to use a credential more often. This is always possible as digital data can be arbitrarily reproduced. For this case we require mechanisms that allow to detect overspending and, if necessary, to obtain an *escrow*. The escrow is certified information about the user that is hidden until an overspending occurs. Only then it can be obtained to reveal for instance the user's identity or her bank-account number.

In addition to enabling applications such as e-cash and e-coupons, restricting the number of times a credential can be shown *in a certain context* is an important security precaution against the sharing and theft of credentials. With context-dependent limited spending we mean that given a concrete context, e.g., a time and place such as "at verifier X on January 1st, 2009", the credential can only be shown a limited number of times in this context. Legitimate anonymous shows from different contexts are however always unlinkable. Applications such as e-cash can be seen as a special case of context-dependent limited spending in which the context is the empty string ϵ.

Technically the limited spending of anonymous credentials is enforced using cryptographic serial numbers. A cryptographic serial number looks like a random number, but is in fact deterministically derived from a unique *seed* in a credential, the *spending context*, and the number of times that the credential has already been shown in this context. This determinism guarantees that for each credential there can only exist up to the *spending limit* many different serial numbers per context. If a user, say Alice, wants to use a credential more often she is forced to reuse one of these serial numbers, which in turn can be detected.

We extend our policy language with a predicate that ensures the correctness of a cryptographic serial number S.

Definition 5.5 (Cryptographic serial numbers). Condition *SerialDer*($S, A,$ *context*, *limit*) refers to S being one of the *limit* valid serial numbers for context *context* and seed A.

Several anonymous credential schemes and related protocols, such as anonymous e-cash realize some form of cryptographic serial numbers (e.g., [15,20,21,30,48,64]).

Cryptographic serial numbers restrict the unlinkability of anonymous credential shows, but a malicious anonymous user can still get away with showing the same serial number multiple times. The server is supposed to maintain a database with spent serial numbers. If the shown number already occurs in the database, then the credential is clearly being overspent, so the server can refuse access.

In some situations however, checking the serial number in real time against a central database is impossible. For example, spending could occur at thousands of servers at the same time, so that the central database would become a bottleneck in the system, or spending could occur offline. In this case, the server cannot refuse access when a credential is being overspent, but needs a way to detect overspending after the fact, and a way to de-anonymize fraudulent users.

Anonymous credentials again offer a solution. When showing a credential, the user can give a piece of (certified) identity information in *escrow*, meaning that this identity information is only revealed when overspending occurs. She does so by at each spending releasing an *escrow share*. If two escrow shares for the same serial number are combined they reveal the embedded identity information, but a single share does not leak any information.[4]

In our policy language, the requirement to give a piece of identity information in escrow is expressed as follows.

Definition 5.6 (Cryptographic escrow). Condition *EscrowShare*(*ess*, S, A, *context*, *limit*, $\langle exp \rangle$) refers to *ess* being a valid escrow share of the attribute expression exp for context *context*, spending limit *limit*, and seed A.

A subset of the anonymous credential schemes and related protocols that support cryptographic serial numbers also support cryptographic escrow, e.g., [20,21,30,48].

Example 5.6. Alice's could receive a gift credential from her rich sister Alicia that Alice can spend on buying 3 expensive wines (but not too expensive). The gift credential is a credential under pk_W, the wine shops public key, containing attribute-value pairs

$$(\texttt{seed} = \ldots, \texttt{maxprice} = \texttt{50EUR}).$$

[4]To avoid that a malicious user reveals the same escrow share twice, escrow shares have to be computed with respect to a unique nonce that is part of the share. The verifier of the anonymous credential is responsible for checking that this value is globally unique. One way of guaranteeing this is to make sure that the nonce is verifier dependent and time dependent. In addition, the verifier can keep a small cache of already used nonces for a certain time interval. Another option is for the verifier to send a random nonce as a challenge before the credential show.

When Alice wants to hand in her gift credential she computes a serial number S and proves the claim

$$\texttt{three\_gift\_credential}^{pk_W}[\texttt{maxprice} \geqslant \mathit{price},\ \mathit{SerialDer}(S, \texttt{seed}, \epsilon, 3)].$$

The wine shop checks that it has not received the same S before to check the validity of the gift certificate.

An escrow based gift credential scheme might be useful if the gift credential should also be accepted by partner shops that might not be constantly online.

6. Policy languages in PRIME

6.1. Scenario

In the PRIME reference scenario (Section 3), our distributed infrastructure includes: *users*, human entities that request on-line services to a *service provider*, which collects personal information before granting an access to its resources, and *external parties* (e.g., business partners) with which a service provider may want to share or trade users' personal information. We assume that the functionalities offered by a service provider are defined by a set of data objects/services. We also assume that, once the personal information is transmitted, the data recipients (i.e., both the service provider and external parties) handle it in accordance with the relevant users' privacy preferences.

When a user needs to access a service, she is required to complete a registration process. Registered users are characterized by a unique *user identifier* (user id, for short). When registration is not mandatory, non-registered users are characterized by a *persistent user identifier* (pseudonym). In this case, personal information is stored under pseudonyms and not users' real identities. Pseudonyms are generated from a master secret each user is supposed to be provided with by means of the *NymDer* algorithm described in Section 5.2.2. Users are given the possibility to link different sessions by using the same pseudonym, or to keep them unlinkable by generating a new pseudonym each time. After this initial set-up phase is completed, what follows is regulated by an access control policy.

Since in open environments the access decision is often based on properties of the user rather than its specific identity, we assume that each party has a *portfolio* of *credentials* issued and certified by trusted authorities (including the party itself). Credentials belong to a partially ordered set, induced by means of an abstraction; for instance, an `identity_document` can be seen as an abstraction for `driver_license`, `passport`, and `identity_card`. Optionally, when a user

shows credentials to a service provider, the relevant information can be stored into a *user profile* associated with the user's identity or one of her pseudonyms. Since an object is not accompanied by any credential, we assume that an *object profile* describing it in the form of a sequence of attribute-value pairs is stored locally at the service provider.

Finally, abstractions can be defined within the domains of users as well as objects. Intuitively, abstractions allow to group together users (objects, resp.) with common characteristics and to refer to the whole group with a name.

6.2. Privacy-aware policies

Several desiderata that privacy-aware policies should satisfy guided our work. One of the major challenges in the definition of a privacy-aware policy language is to provide *expressiveness* and *flexibility* while at the same time ensuring *ease of use* and therefore *applicability*. A privacy-aware policy should then be based on a high level formulation of the rules, possibly close to natural language formulation. The definition of generic conditions based on *context* information should be supported, including location information [1], to allow environmental factors to influence how and when the policy is enforced. Moreover, the policy definition should be fully integrated with subject and object *ontologies* in defining access control restrictions. Also, privacy-aware policies should take advantage of the integration with credentials ontology that represents relationships among attributes and credentials. In addition to traditional server-side access control rules, users should be able to specify *client-side restrictions* on how the released information can be used by their remote counterpart. As both the server may not have all the needed information for an access grant decision and the user may not know which information she needs to present to a (possibly previously unknown) server, an *interactive* way of enforcing the access control process is required.

In the following, we introduce different types of policies based on terms and predicates introduced in Section 5 and summarized by the grammar in Fig. 3.

6.2.1. Access control policies

Access control policies (ACP) regulate access to Personal Identifiable Information (PII), data objects and services (i.e., objects). They define positive authorization rules, which specify a set of *conditions* to be satisfied by a *subject* to perform a specific *action* on an *object*. In the literature, access control policies that protect PII may be referred to as *release policies* [12].

Basic elements of the language. The set of basic literals used in the access control policy definition includes the building blocks described in Fig. 3 and a set of domain-dependent predicates. To refer to the user (i.e., the subject) and the target (i.e., the object) of the request being evaluated without the need of introducing variables in the language, we use keywords **user** and **object**, respectively, whose occurrences in

a claim are intended to be substituted by actual request parameters during run-time evaluation of the access control policy.[5]

We have identified three main basic elements of the language: *subject_claim*, *object_claim* and *conditions*.

Subject claims: These claims allow for the reference to a set of subjects depending on whether they satisfy given conditions that can be evaluated on the subject's profile. More precisely, a subject claim is a $\langle claim \rangle$ as defined in Fig. 3. The following are examples of subject claims:

- `identity_card`^{pk_1}[`maritalStatus` =‘married’, `nationality` = ‘EU’] denotes European users who are married. These properties should be certified by showing the `identity_card` credential verifiable with public key pk_1.
- `identity_card`^{pk_1}[`age` $<$ *25*] denotes users with age less than 25. This property can potentially be certified by showing an anonymous `identity_card` credential, verifiable with public key pk_1.
- *VerEnc*(C, pk_1, `identity_card`^{pk_2}[`name`, `address`], “*disputes*”) requests the release of attributes `name` and `address`, possibly, in the form of a ciphertext C encrypted under public encryption key pk_1. These attributes will be decrypted by a trusted third party only in cases of *disputes*.

Object claims: These claims refer to a set of objects depending on whether they satisfy given conditions that can be evaluated on the objects' profile. Objects' attributes are referenced through the usual dot notation **object**.`AttributeName`, where **object** uniquely identifies the object at run-time evaluation, and `AttributeName` is the name of the property. More precisely, an *object claim* is a positive boolean combination of formulae of the form A_i *math* a_i. For example, the claim “**object**.`expiration` $>$ `today`” denotes all objects not yet expired.

Conditions: A *conditions* element specifies restrictions that can be satisfied at run-time while processing a request. *Conditions* are boolean formulae in the form of `predicate_name`(*arguments*), where `predicate_name` belongs to a set of domain-dependent predicates dealing with: (i) trust-based conditions, (ii) location-based conditions [1]; and (iii) other conditions regarding the information stored at the server. *Arguments* is a list, possibly empty, of constants or attributes.

Policy and rule definition. Syntactically, access control policies are composed by a set of authorization rules defined as follows.

[5] We adopt the keyword **user** to make our solution compatible with other approaches that allow for conditions based on uncertified statements (e.g., *declarations* in [12]).

Definition 6.1 (Access control rule). An access control rule is an expression of the form ⟨*subject*⟩ [WITH ⟨*subject_claim*⟩] CAN ⟨*actions*⟩ ON ⟨*object*⟩ [WITH ⟨*object_claim*⟩] FOR ⟨*purposes*⟩ [IF ⟨*conditions*⟩].

An access control rule defines for which actions and purposes,[6] a user identified by the pair ⟨*subject*⟩ (i.e., a user identifier or a named abstraction) and ⟨*subject_claim*⟩ can access an object identified by the pair ⟨*object*⟩ (i.e., an object identifier or a named abstraction) and ⟨*object_claim*⟩. Also, the rule defines the conditions (i.e., ⟨*conditions*⟩ element) to be satisfied before any access is granted.

6.2.2. Data handling policies

Building up a policy in accordance to the users' privacy preferences is far from simple a task. We have to tackle the trade-off between simplicity and expressiveness to at least ensure individual control, consent, modifiability, and data security [51]. To fulfill these requirements, personal information collected for one purpose must not be used for any other purpose unless an explicit consent has been provided by the relevant user. A *data handling policy* (DHP) [2] enables a user to define how her PII can be used by the service provider and/or external parties. In our approach, DHP are *sticky*, that is to say, they physically follow the data during the release to an external party, thus allowing for a chain of control starting from the data owner.

In a DHP specification, two main issues must be dealt with: *by whom* and *how* a policy is defined. There are several possibilities to the former problem, ranging from *server-side* to *user-side* solutions, each of them requesting a specific level of negotiation. In this work, we adopt a balanced approach, where predefined policy templates are provided by the service provider to the user at the moment of a data request. The templates are then customized to meet different privacy requirements of each user. The customization process may be entirely led by the user, or some suggestions may be proposed by the service provider. A DHP is agreed upon when the customized template is accepted by the service provider. This represents a very flexible strategy for the definition of data handling policies and a good trade-off between the power given to the service providers and the protection assured to the users.

With respect to the latter issue (i.e., how a DHP is defined), DHP are expressed as independent rules and represent the user's privacy preferences on how external parties can use her personal data. Personal data are then *tagged* with such DHP. This approach provides a good separation between ACP and DHP that have two distinguished purposes. Such separation makes DHP more intuitive and user-friendly, reduces the risk of having unprotected data types and, finally, makes easier the customization of additional components such as recipients and actions.

[6] We suppose that actions and purposes are defined in suitable domain-dependent ontologies.

Basic elements of the language. The basic elements of a DHP are: *recipients*, *purposes*, *PII abstraction*, and *restrictions*.

Recipients: A recipient is an external party which can get access to PII [32]. Since external parties may be unknown to the user, the set of entities to which her data may be disclosed must be set without information on their identity. A PII recipient is determined on the basis of her attributes which must satisfy a specific set of conditions, as for the ACP's *subject_claim* in Section 6.2.1. Conditions (as discussed in Section 6.1) are evaluated on the credentials of the recipient.

Purposes: They identify the objectives for which the information can be used. The domain of purposes can be structured by means of abstractions in the form of generalization/specialization relationships that group together those purposes showing common characteristics.

PII abstraction: *Data types* can be introduced as abstractions of PII to allow for the expression of DHP in terms of data types, rather than single properties of a user. A hierarchy of data types can also be built.

Restrictions: Restrictions collect conditions that must be satisfied before or after access to personal data is granted. We distinguish between *provisions*, *obligations*, and *generic* conditions which are optional boolean combinations of formulae in the form of `predicate_name`(*arguments*), where `predicate_name` belongs to a set of domain-dependent predicates, and *arguments* is a list, possibly empty, of constants or variables on which predicate `predicate_name` is evaluated. More in detail:

- *provisions* represent actions that must be performed before an access can be granted [10]. For instance, a business partner can read the email addresses of a user provided that it has *paid a subscription fee*;
- *obligations* represent actions that have to be either performed immediately after an access has been granted [10] or at a later time, when specific events occur (e.g., time-based or context-based events [16]). For instance, a data retention restriction may be imposed on how long personal data should be retained (e.g., `delete_after`(*num_days*));
- *generic* conditions either evaluate properties of users' profiles, like membership in specific groups, or represent conditions that can be satisfied at run-time when a request is processed. For instance, `access_time`(8*am*, 5*pm*) is satisfied if the access request is sent between 8am and 5pm.

Policy and rule definition. Syntactically, a DHP has the form "⟨*PII*⟩ MANAGEDBY ⟨*DHP_rules*⟩", where *PII* identifies a PII abstraction and *DHP_rules* identifies one or more rules, composed in OR logic, regulating the access to the PII data to which they refer. In a DHP template, the *PII* element represents the name of an attribute or the name of a data type. When it is part of a customized DHP, it represents an attribute belonging to a privacy profile. Formally, a DHP rule can be defined as follows.

Definition 6.2 (DHP rule). A *DHP rule* is an expression of the form ⟨*recipients*⟩ CAN ⟨*actions*⟩ FOR ⟨*purposes*⟩ [IF ⟨*gen_conditions*⟩] [PROVIDED ⟨*prov*⟩] [FOLLOW ⟨*obl*⟩].

A DHP rule specifies that *recipients* can execute *actions* on *PII* for *purposes* provided that *prov* and *gen_conditions* are satisfied, and with obligations *obl*.

6.3. Regulating the dialog between parties

Policies dealing with PII may be considered sensitive data themselves, whose transmission has to be carefully considered [61,75]. The dialog between the parties should then be regulated, by providing filtering functionalities that limit the release of sensitive information related to the policy itself. The partial disclosure of policies affects also the use of credentials, as shown in the following discussion. Whenever a condition in the form of $\texttt{cred\_type}^{pk}[A\ math\ a]$ is to be disclosed to a user as part of a service provider's ACP, the latter has three options.

Minimal policy disclosure prescribes the most restricted presentation of a condition in the policy, such that only the attribute name of the triple is presented to the user. This approach equates to requesting the value of the attribute A without revealing how this information will be evaluated with respect to the ACP (i.e., $\texttt{cred\_type}^{pk}[A\ -\ -]$, where $-$ works as a placeholder for hidden predicates and values). The request will be met by a positive response if the user's trust in the service provider is high enough to allow for the attribute to be sent without further information on the relevant ACP. The attribute value is then sent to the service provider. Otherwise, the request is turned down.

Partial policy disclosure prescribes a presentation of the attribute name and the predicate in the ACP condition. The user is presented with a partially hidden condition like $\texttt{cred\_type}^{pk}[A\,math\,-]$. In this case the user has two ways to provide a positive response: the value of the A attribute can be revealed in full and sent as in the minimal disclosure case (mandatory option when the predicate is '='), or the user can use an anonymous credential and prove that her attribute A fulfills a given condition based on *math* and a value k. For instance, when presented with $\texttt{cred\_type}^{pk}[A \geqslant -]$, the user can respond by providing a certified proof that $A \geqslant k$. If k is greater than or equal to the actual value in the ACP, the service provider will consider the condition fulfilled, otherwise it will issue the same request once again, and it will be up to the user to disclose the attribute value or to provide another proof with a k' greater than k.

Full policy disclosure is obtained when the service provider presents the $\texttt{cred\_type}^{pk}[A\,math\,a]$ condition in full to the user, who has the choice to reveal the attribute value, or to provide an anonymous credential proving

that the condition in the ACP is actually met. Again, in case the condition presented to the user contains a predicate '=', the user is requested to present the exact attribute value. Nevertheless, in this case the user is given more information about the service provider's ACP and she can decide to proceed with the sending of the data only when a successful outcome is possible.

6.4. Policy negotiation and evaluation

The PRIME reference scenario is aimed at supporting two different interactions between the parties: the *User-Service Provider interplay* (see Fig. 4(a)), which is carried out when a user submits an access request for a resource managed by the service provider, and the *External Party-Service Provider interplay* (see Fig. 4(b)), which can take place at a later stage, when an external party submits an access request for PII of the user stored by the service provider.[7] The access request submitted by a user or an external party can be defined as follows.

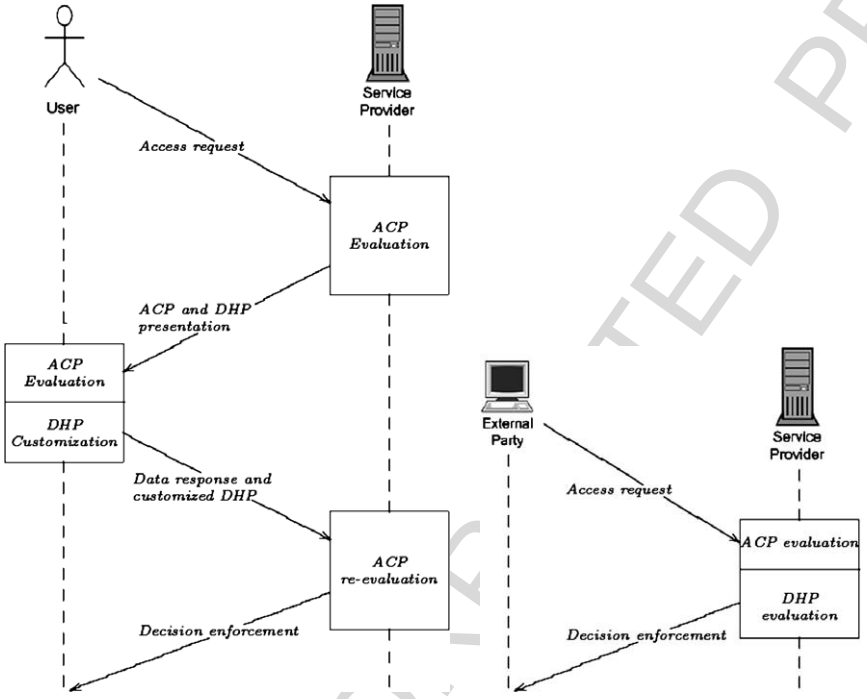

Fig. 4. User-Service Provider interplay (a) and External Party-Service Provider interplay (b).

[7] For the sake of simplicity, Fig. 4(b) does not repeat the intermediate steps showed in the user-service provider interplay.

Definition 6.3 (Access request). An *access request* is a tuple of the form ⟨*user_id*, *action*, *object*, *purposes*⟩, where *user_id* is the identifier/pseudonym of the user, *action* is the action that is being requested, *object* is the object on which the user wishes to perform the action, and *purposes* is the purpose or a group thereof for which the object is requested.

In the following, we concentrate our discussion on the phases composing a generic interplay between the parties, originated by an access request and resulting in a service release.

Phase 1: Access request. Each interplay between a user and a service provider begins with a request in the form of ⟨*user_id*, *action*, *object*, *purposes*⟩. In this scenario, the *object* can consist of either a service provided by the service provider or a PII collected by the service provider during previous transactions. The access request is then evaluated as illustrated in the following.

Phase 2: ACP evaluation (service provider side). The access request is evaluated against the applicable access control rules. The set of applicable access control rules includes those rules whose *actions*, *purposes*, and *object* include the relevant items specified in the access request and the *object* of the access request satisfies the *object_claim* in the access control rules. The default access decision is "no" (*deny-all*), that is, if no rule is applicable, the access is denied. The conditions (i.e., *subject_claim* and *conditions*) in the applicable rules are evaluated. A "yes", "no", or "undefined" access decision is obtained at the end of the evaluation phase. In case of a negative ("no") access decision, the process terminates. An "undefined" access decision means that the information provided by the user is not sufficient to determine whether the request can be granted or denied. Additional information is required by means of filtered queries to the user, so that disclosure of sensitive information related to the policy itself is avoided (see *Phase* 3 and Section 6.3). Finally, in case of a positive ("yes") access decision, that is, there exists at least one rule such that *subject_claim* and *conditions* evaluate to true based on the user's profile, the access control evaluation ends successfully and the system gets on to verify whether there exists some restrictions on the secondary use of the requested target (see *Phase* 6). As said, since the service provider may not have all the needed information for an access grant decision, an *interactive* way of enforcing the access control process is required. In this phase, a *partial evaluation* approach is used meaning that the service provider evaluates those conditions for which data are available and interacts with the counterpart to evaluate the remaining ones. For instance, suppose that the *subject_claim* of an applicable rule contains "$\texttt{Age} > 18 \wedge \texttt{nationality} = \text{`}EU\text{'}$" and that *conditions* is empty. Three cases can happen: (i) if the service provider knows that the user is greater than 18 and European, the *subject_claim* is evaluated to true, the rule is then satisfied, and the evaluation process gets to evaluate the relevant DHP (see *Phase* 6); (ii) if the service provider knows that the user is *European*,

the *subject_claim* is evaluated to undefined, and the service provider communicates with the user to evaluate condition "Age $>$ 18" (see *Phase* 3); (iii) if the service provider knows that the user is less than 18, the *subject_claim* is evaluated to false, and the process aborts.

Phase 3: ACP and DHP presentation. This phase focuses on those conditions not yet evaluated to true nor false due to lack of user information, which must then be presented to the user. Before being sent, conditions are possibly processed to meet the required ACP disclosure level, and relevant DHP templates are attached. In the case of a *partial policy disclosure*, several request/response message pairs may be exchanged between the user and the service provider before an agreement is reached.

Phase 4: ACP evaluation (user side) and DHP customization. After receiving the request for information with the relevant DHP templates, the user selects her applicable access control policies as in *Phase* 2. Based on the applicable policies evaluation, the user identifies the credentials she is willing to release to the service provider. If the DHP templates can be customized to meet the user's preferences, the data can be released, and the customized templates are sent along. In general, the data release process could require multiple negotiation steps [74]. A straightforward extension to our solution would take into account situations where the user requires the service provider to release some PII as well, for which a specific DHP is defined.

Phase 5: ACP re-evaluation (service provider side). When the service provider receives the requested data together with the customized DHP, it re-evaluates the access request against the applicable policies selected in *Phase* 2. If the evaluation result is "yes", the process continues with *Phase* 6; otherwise the process aborts.

Phase 6: DHP evaluation (external party-service provider interplay only). The DHP attached to the *object* of the request are evaluated by first selecting the applicable rules. The set of applicable data handling rules contains those rules for which their *actions* and *purposes* include the *action* and *purposes* specified in the access request, respectively. For each applicable data handling rule, all the conditions expressed in the *recipients*, *gen_conditions*, and *prov* fields are evaluated. At the end of this evaluation phase, a "yes", "no", or "undefined" access decision is reached, and it is managed as described in *Phases* 2–4. The only difference is that no policy hiding is performed here. In particular, in case of a positive access decision, that is, there exists a rule such that *recipients*, *gen_conditions*, and *prov* evaluate to true, the access is granted and the evaluation process is completed in *Phase* 7. For instance, suppose that a DHP states that business partners of *CyberWinery* (recipient) can read (action) the emails of a user (PII) for service release (purpose) with the obligation of deleting the data after 30 days. If a request in the form $\langle uid, read, email, service\_release \rangle$ is submitted and the user is a business partner of *CyberWinery*, the data handling rule is evaluated to true.

Phase 7: Decision enforcement. This phase consists of the enforcement of the final access control decision. Access is granted, if at least one ACP of the service provider and one DHP attached to the requested data are evaluated to true. In such circumstances, the requested PII/data object/service is released together with the corresponding DHP. The party is then responsible for managing the received data/service in accordance with the attached DHP. Moreover, upon the receipt, the relevant obligations inside the DHP must be enforced. Let us take up the example in *Phase* 6: the obligation field states `delete_after_time`(30 days). Thus, as soon as the 30 days are expired, the data must be deleted by the external party.

6.5. The CyberWinery use case

Based on the scenario depicted in Section 3 and on the interplay in Fig. 4, we provide an example of a full interplay involving three major parties in the `CyberWinery` scenario: `Alice`, `CyberWinery`, and `LogisticsProvider`. `Alice` wants to buy a precious bottle of Italian red wine at `CyberWinery`'s website. To this aim, she submits a request in the form $\langle$*Alice*, *execute*, *buy@CyberWinery*, *personal purchase*$\rangle$.

The request is evaluated by the `CyberWinery` against the ACP in Table 1. Based on *action*, *object*, and *purpose* in the request, ACP1 is the only applicable policy

Table 1

An example of access control policies (ACP1 and ACP2) of `CyberWinery` and an access control policy (RP1) of `Alice`

	AC Rules	Description
ACP1	any WITH (`identity_card`^{pk_1}`[age` $>$ 18, `nationality` $\in$ EU] $\vee$ `identity_card`^{pk_1}`[age` $>$ 21, `nationality` $\in$ non-EU]) $\wedge$ *VerEnc*(C, pk_{s1}, `identity_card`^{pk_1}`[address]`, "shipping") $\wedge$ `credit_card`^{pk_2}`[number, circuit, expiration]` $\wedge$ *VerEnc*(C, pk_{s2}, `credit_card`^{pk_2}`[name]`, "failedpayment") $\wedge$ (`identity_card`^{pk_1}`[name]` $=$ `credit_card`^{pk_2}`[name]`)) CAN `execute` ON `buy@CiberWinery` FOR `personal purchase`	A user is authorized to execute `buy@CyberWinery` service for personal purchase purpose, if she owns a valid credit card, and she releases the identity card address, the credit card number, circuit, expiration, and name (name possibly encrypted), and she is European and older than 18 or she is non European and older than 21.
ACP2	any WITH `identity_card`^{pk_1}`[age` $>$ 16] CAN `browse` ON `CyberWinerySite` FOR `window shopping` IF `log_access()`	A user older than 16 can browse the `CyberWinery` Web site for window shopping purposes, if access is logged.
RP1	any WITH `business_card`^{pk_b}`[BBB_certified` $=$ "yes"] CAN `access` ON `cc_info` WITH **object**`.expiration` $>$ *today* FOR `complete purchase`	A user is willing to give access to her valid credit card information only to BBB-certified entities for a complete purchase purpose.

and the access request is evaluated against it. Let us suppose this is the first request by Alice, and then she is unknown to CyberWinery (i.e., her profile at CyberWinery is empty). The access evaluation result is "undefined" and Alice is prompted by CyberWinery with a request for additional information together with applicable DHP templates. For the sake of clarity, we assume that the dialog is regulated by the full policy disclosure approach. CyberWinery asks Alice for the following set of information: (i) a certification of the fact that she is European and greater than 18 or non-European and greater than 21; (ii) a proof of possession of a credit card and the release of some attribute values in it, that is, *number*, *circuit*, *expiration*, *name* (attribute *name* can be released by means of a verifiable encryption); (iii) a verifiable encryption containing the *address* to be used in the shipping process.

After receiving the request for information, Alice selects her applicable access control policies (RP1 in Table 1). Based on RP1, Alice is willing to release the data requested by CyberWinery if CyberWinery proves its membership to the BBB. If this condition is verified, Alice customizes the received DHP templates and releases data together with the customized DHP (see Table 2).

As soon as CyberWinery receives Alice's data, it re-evaluates the access request against ACP1. Let us suppose Alice releases all the requested information

Table 2

An example of customized data handling policies that protect Alice's data stored by CyberWinery

	Data Handling Policies		
	PII	DHP Rules	Description
DHP1	Alice.cc_info	business_card^{pk_3}[company = 'CyberWinery'] CAN read FOR complete purchase PROVIDED log_access() FOLLOW delete_after(purchase satisfied)	An employee of CyberWinery can read the cc_info of Alice for complete purchase purposes provided that the access is logged. The data must be deleted after purchase is completed.
DHP2	Alice.address	business_card^{pk_3}[company = 'Logistics-Provider'] CAN decrypt FOR shipping FOLLOW notify(Alice)	An employee of LogisticsProvider can decrypt the address of Alice for shipping purposes. Data decryption must be notified to Alice.
DHP3	Alice.name	business_card^{pk_3}[company = 'CyberWinery'] CAN decrypt FOR dispute resolution PROVIDED log_access() FOLLOW delete_after(6 months)	An employee of CyberWinery can decrypt the name of Alice for dispute resolution purposes provided that the access is logged. Data must be deleted after six months.

and that she is 25 years old and European. ACP1 evaluates to true, the access to the `buy@CyberWinery` service is granted, and `Alice` buys the bottle of wine she wanted.

To complete the purchase process, `CyberWinery` needs to contact an external party, called `LogisticsProvider`, responsible for the shipping process. To send the wine to `Alice`, `LogisticsProvider` needs to decrypt the address information of `Alice`. Before any access is given to `Alice`'s data, the DHP in Table 2 must be evaluated against `LogisticsProvider`'s request (i.e., ⟨*LogisticsProvider, decrypt, Alice.address, shipping*⟩).[8] The only applicable policy is DHP2, which evaluates to true. `LogisticsProvider` then decrypts the address information, sends the bottle of wine to `Alice`, and the relevant obligations are enforced. In our case, according to the obligations in DHP2, `Alice` must be notified about the access to her address data.

7. Related work

A number of projects and research works about privacy and identity management have been presented in the last few years, although not many of them have addressed the issue of exploiting cryptography for the definition of a privacy-enhanced access control. Three lines of research are closely related to the topics of this paper: (i) the definition and development of credential-based access control models and trust negotiation solutions, (ii) the definition and development of access control and privacy-aware languages, and (iii) the definition of anonymous credentials.

Access control models exploiting digital credentials make access decisions on whether or not a party may execute an access on the basis of properties that the requesting party may have. Traditional access control solutions [12,43,45,50,75], which exploits properties proven by one or more certificates, are focused on providing expressive and powerful logic languages and do not consider privacy of the users as a primary design requirement. The first proposals that investigate the application of credential-based access control regulating access to a server are done by Winslett et al. [60,69]. Access control rules are expressed in a logic language, and rules applicable to a service access can be communicated by the server to the clients. Bonatti and Samarati provide a first attempt to build a uniform framework for attribute-based access control specification and enforcement [12]. Access rules are specified in the form of logical rules, based on a formal language that includes some domain-specific predicates. Attribute certificates are modeled as credential expressions. Moreover, this work introduces a type of unsigned statements, namely declarations, that together with properly specified user profiles aim at enabling a server to reach an access decision in the absence of certified credentials. In the proposed framework, the communication of requisites a requester must satisfy is based

[8] Also in this case ACP of `CyberWinery` must be evaluated. For sake of conciseness, both in Table 2 and in the discussion these additional ACP are not described.

on a filtering and renaming process applied to the server's policies, exploiting partial evaluation techniques traditionally associated with logic programs. Differently from the above approaches, the work in this paper is focused on the definition of a privacy-enhanced access control system that includes different models and languages. The presented infrastructure is then aimed, on the one side, to regulate access to resources and, on the other side, to protect the privacy of the users. A major requirement considered in our work, and neglected by current solutions, is the integration of the policy languages with anonymous credentials definition and evaluation.

Several automated trust negotiation proposals have been developed [61,73,74]. A gradual trust establishment is obtained by requesting and consequently disclosing credentials [37]. In [59,66,72,73,75], trust negotiation issues and strategies, which a user can apply to select credentials to submit to the opponent party during a negotiation, are investigated. Our work is not aimed to develop another complex negotiation protocol; rather, our approach focuses on providing a privacy-enhanced access control infrastructure, whose fundamental requirements are ease of use and applicability from a user perspective. Our work is complementary to existing trust negotiation solutions and could be applied in conjunction with them towards the development of a complete framework addressing different aspects of the privacy problem.

Recently, several access control [3,34,67] and data handling languages [4,6,70] have been defined, and some of them have provided preliminary solutions to the privacy issue. eXtensible Access Control Markup Language (XACML) [34], an OASIS standardization effort, proposes a XML-based language to express and interchange access control policies. In addition to the language, also an architecture for the evaluation of policies and a communication protocol for messages exchange are defined as part of the proposal. Ardagna et al. [3] present a privacy-enhanced authorization model and language for the definition and enforcement of access restrictions based on subjects' and objects' properties. They also suggest a way to exploit the Semantic Web to allow for the definition of access control rules based on generic assertions that are expressed on the basis of ontologies that control metadata content. These rules are then enforced on resources tagged with metadata defined by the same ontologies. The W3C consortium proposed the Platform for Privacy Preferences Project (P3P) [28, 70] that tackles the need of a user for assessing whether a service provider's privacy policy complies with her privacy requirements. P3P provides a XML-based language and a mechanism to ensure that users release personal information only after being properly informed about the relevant data treatment. A P3P Preference Exchange Language (APPEL) [71] enables users to specify their privacy preferences. APPEL can be used by users' agents to make automated or semi-automated decisions about the machine-readable privacy policies of P3P-enabled Web sites. Enterprise Privacy Authorization Language (EPAL) [5,6] is a XML-based markup language and architecture for formalizing, defining, and enforcing enterprise-internal privacy policies. It addresses the problem on the server side and supports a company in the tasks of specifying access control policies, with reference to attributes/properties of requesters

and protecting users' private information. EPAL aims at enabling organizations to translate their privacy policies (possibly written in P3P) into IT control statements and to enforce them. In general, these languages mainly fail in providing a complete and comprehensive solution that allows the users to access services still protecting their privacy. For instance, XACML provides an expressive attribute-based access control language, but does not protect users' privacy. P3P, instead, provides a language for regulating secondary uses of data based on users' preferences, but it is based on categories only, does not rely on credentials, and supports "all or nothing" approach making the overall privacy protection weak. By contrast, the infrastructure in this paper is a privacy-oriented solution where access control and data handling mechanisms are integrated with anonymous credentials in a comprehensive framework. Therefore, users can access a service still protecting their personal information and gaining a level of control over their information.

The basic principle of anonymous credentials was put forward by Chaum [17, 18], and the first, albeit rather inefficient scheme is due to Damgård [29]. More efficient schemes were later proposed by Brands [13,14] and by Camenisch and Lysyanskaya [22–24,47]. The scheme from [22] has been implemented in the Idemix credential system [8,26,42], and was also used in the Direct Anonymous Attestation protocol [15] in the Trusted Platform Module specification of the Trusted Computing Group [65].

The type of verifiable encryption mentioned in Section 5 was independently proposed in [7,19]; the most efficient scheme currently known is due to Camenisch and Shoup [25]. Several techniques to limit the number of times credentials can be shown (or spent) have appeared in the literature [15,20,21].

8. Conclusion

The PRIME project has shown that privacy-enhancing identity management is feasible from a technical point of view. Although we currently see that the industry embraces some of these concepts, there is a lot of work to do make PRIME's vision an every day reality. First, today's infrastructure must be change to employ the privacy-enhancing technologies discussed in this paper which in turn requires that standards on many levels are worked out. Then, there are still many open research problems to be solved, ranging from cryptographic research, to policy languages, and, probably most importantly, interfaces that allow users to manage their identities in an intuitive way.

Luckily, we see lots of efforts world wide to solve all these problems. To name just a few, there efforts include EU-funded projects such as PrimeLife, Picos, Swift, and TAS3; standardizations by W3C, OASIS, ISO, and ITU; and many scientific communities as witnessed by numerous conferences and workshops.

Acknowledgements

This paper reports only a small fraction of the results achieved by PRIME. All the numerous people working on PRIME contributed in one or the other way to the presented result. Thanks to all of you for the many inspiring and charming discussions!

The research leading to these results has received funding from the European Community's Sixth Framework Programme through project PRIME (IST-2002-507591) and from the Seventh Framework Programme through project PrimeLife (grant agreement no. 216483).

References

[1] C.A. Ardagna, M. Cremonini, E. Damiani, S. De Capitani di Vimercati and P. Samarati, Supporting location-based conditions in access control policies, in: *Proc. of the ACM Symposium on Information, Computer and Communications Security (ASIACCS'06)*, Taipei, Taiwan, March 2006.

[2] C.A. Ardagna, M. Cremonini, S. De Capitani di Vimercati and P. Samarati, A privacy-aware access control system, *Journal of Computer Security (JCS)* **16**(4) (2008), 369–392.

[3] C.A. Ardagna, E. Damiani, S. De Capitani di Vimercati and P. Samarati, Towards privacy-enhanced authorization policies and languages, in: *Proc. of the 19th Annual IFIP WG 11.3 Working Conference on Data and Applications Security*, Storrs, CA, USA, August 2005.

[4] G.-J. Ahn and J. Lam, Managing privacy preferences in federated identity management, in: *Proc. of the ACM Workshop on Digital Identity Management*, Fairfax, VA, USA, November 2005.

[5] P. Ashley, S. Hada, G. Karjoth, C. Powers and M. Schunter, Enterprise privacy authorization language (EPAL 1.1), Technical report, IBM Research, 2003; http://www.zurich.ibm.com/security/enterprise-privacy/epal.

[6] P. Ashley, S. Hada, G. Karjoth and M. Schunter, E-P3P privacy policies and privacy authorization, in: *Proc. of the ACM workshop on Privacy in the Electronic Society (WPES 2002)*, Washington, DC, USA, November 2002.

[7] N. Asokan, V. Shoup and M. Waidner, Optimistic fair exchange of digital signatures, *IEEE Journal of Selected Areas in Communications* **18**(4) (2000), 591–610.

[8] M. Backes, J. Camenisch and D. Sommer, Anonymous yet accountable access control, in: *Proc. of the 2005 ACM Workshop on Privacy in the Electronic Society*, Alexandria, VA, USA, November 2005, pp. 40–46.

[9] O. Berthold, H. Federrath and S. Köpsell, Web MIXes: A system for anonymous and unobservable Internet access, in: *Proceedings of Designing Privacy Enhancing Technologies: Workshop on Design Issues in Anonymity and Unobservability*, H. Federrath, ed., Lecture Notes in Computer Science, vol. 2009, Springer, 2000, pp. 115–129.

[10] C. Bettini, S. Jajodia, X. Sean Wang and D. Wijesekera, Provisions and obligations in policy management and security applications, in: *Proc. of the 28th VLDB Conference*, Hong Kong, China, August 2002.

[11] J.-F. Blanchette and D.G. Johnson, Data retention and the panoptic society: The social benefits of forgetfulness, *The Information Society* **18** (2002), 33–45.

[12] P.A. Bonatti and P. Samarati, A unified framework for regulating access and information release on the web, *Journal of Computer Security* **10**(3) (2002), 241–272.

[13] S. Brands, Restrictive blinding of secret-key certificates, Technical Report CSR9509, CWI Amsterdam, 1995.

[14] S. Brands, Rethinking public key infrastructure and digital certificates – building in privacy, PhD thesis, Technical University Eindhoven, 1999.

[15] E.F. Brickell, J. Camenisch and L. Chen, Direct anonymous attestation, in: *11th ACM Conference on Computer and Communications Security, CCS 2004*, ACM, 2004, pp. 132–145.

[16] M. Casassa Mont and F. Beato, On parametric obligation policies: Enabling privacy-aware information lifecycle management in enterprises, in: *Proc. of the 8th IEEE Workshop on Policies for Distributed Systems and Networks (Policy 2007)*, Bologna, Italy, June 2007.

[17] D. Chaum, Security without identification: Transaction systems to make big brother obsolete, *Communications of the ACM* **28**(10) (1985), 1030–1044.

[18] D. Chaum and J.-H. Evertse, A secure and privacy-protecting protocol for transmitting personal information between organizations, in: *Advances in Cryptology – CRYPTO'86*, A.M. Odlyzko, ed., Lecture Notes in Computer Science, vol. 263, Springer, 1987, pp. 118–167.

[19] J. Camenisch and I. Damgård, Verifiable encryption, group encryption, and their applications to separable group signatures and signature sharing schemes, in: *Advances in Cryptology – ASIACRYPT 2000*, T. Okamoto, ed., Lecture Notes in Computer Science, vol. 1976, Springer, 2000, pp. 331–345.

[20] J. Camenisch, S. Hohenberger, M. Kohlweiss, A. Lysyanskaya and M. Meyerovich, How to win the clonewars: efficient periodic n-times anonymous authentication, in: *13th ACM Conference on Computer and Communications Security, CCS 2006*, A. Juels, R.N. Wright and S. De Capitani di Vimercati, eds, ACM, 2006, pp. 201–210.

[21] J. Camenisch, S. Hohenberger and A. Lysyanskaya, Compact e-cash, in: *Advances in Cryptology – EUROCRYPT 2005*, R. Cramer, ed., Lecture Notes in Computer Science, vol. 3494, Springer, 2005, pp. 302–321.

[22] J. Camenisch and A. Lysyanskaya, An efficient system for non-transferable anonymous credentials with optional anonymity revocation, in: *Advances in Cryptology – EUROCRYPT 2001*, B. Pfitzmann, ed., Lecture Notes in Computer Science, vol. 2045, Springer, 2001, pp. 93–118.

[23] J. Camenisch and A. Lysyanskaya, A signature scheme with efficient protocols, in: *Security in Communication Networks, Third International Conference*, S. Cimato, C. Galdi and G. Persiano, eds, Lecture Notes in Computer Science, vol. 2576, Springer, 2003, pp. 268–289.

[24] J. Camenisch and A. Lysyanskaya, Signature schemes and anonymous credentials from bilinear maps, in: *Advances in Cryptology – CRYPTO 2004*, M.K. Franklin, ed., Lecture Notes in Computer Science, vol. 3152, Springer, 2004, pp. 56–72.

[25] J. Camenisch and V. Shoup, Practical verifiable encryption and decryption of discrete logarithms, in: *Advances in Cryptology – CRYPTO 2003*, D. Boneh, ed., Lecture Notes in Computer Science, vol. 2729, Springer, 2003, pp. 126–144.

[26] J. Camenisch and E. Van Herreweghen, Design and implementation of the Idemix anonymous credential system, in: *9th ACM Conference on Computer and Communications Security, CCS 2002*, V. Atluri, ed., ACM, 2002, pp. 21–30.

[27] R. Clarke, The digital persona and its application to data surveillance, *The information society* **10** (1994), 77–92.

[28] L.F. Cranor, *Web Privacy with P3P*, O'Reilly & Associates, 2002.

[29] I. Damgård, Payment systems and credential mechanisms with provable security against abuse by individuals, in: *Advances in Cryptology – CRYPTO'88*, S. Goldwasser, ed., Lecture Notes in Computer Science, vol. 403, Springer, 1990, pp. 328–335.

[30] I. Damgård, K. Dupont and M.Ø. Pedersen, Unclonable group identification, in: *EUROCRYPT*, S. Vaudenay, ed., Lecture Notes in Computer Science, vol. 4004, Springer, 2006, pp. 555–572.

[31] R. Dhamija and L. Dusseault, The seven flaws of identity management: Usability and security challenges, *IEEE Security and Privacy* **6**(2) (2008), 24–29.

[32] Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data, *Official Journal L 281*, 23/11/1995, pp. 31–50.

[33] R. Dingledine, N. Mathewson and P.F. Syverson, Tor: The second-generation onion router, in: *USENIX Security Symposium*, USENIX, 2004, pp. 303–320.

[34] eXtensible Access Control Markup Language (XACML) Version 2.0, February 2005; http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.pdf.
[35] C. Fried, Privacy, *The Yale Law Journal* **77** (1968), 475–493.
[36] O.H. Gandy, *The Panoptic Sort. A Political Economy of Personal Information*, Critical Studies in Communication and in the Cultural Industries, Westview Press, Boulder, San Francisco, Oxford, 1993.
[37] R. Gavriloaie, W. Nejdl, D. Olmedilla, K. Seamons and M. Winslett, No registration needed: How to use declarative policies and negotiation to access sensitive resources on the semantic web, in: *Proc. of the 1st First European Semantic Web Symposium*, Heraklion, Greece, May 2004.
[38] E. Goffman, *The Presentation of Self in Everyday Life*, Doubleday Anchor Books, Garden City, New York, 1959.
[39] O. Goldreich, S. Micali and A. Wigderson, How to prove all NP statements in zero-knowledge and a methodology of cryptographic protocol design, in: *Advances in Cryptology – CRYPTO'86*, A.M. Odlyzko, ed., Lecture Notes in Computer Science, vol. 263, Springer-Verlag, 1987, pp. 171–185.
[40] S. Goldwasser, S. Micali and R. Rivest, A digital signature scheme secure against adaptive chosen-message attacks, *SIAM Journal on Computing* **17**(2) (1988), 281–308.
[41] M. Hildebrandt and S. Gutwirth (eds), *Profiling the European Citizen*, Springer, 2008.
[42] IDEntity MIXer (IDEMIX); http://www.zurich.ibm.com/security/idemix/.
[43] K. Irwin and T. Yu, Preventing attribute information leakage in automated trust negotiation, in: *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
[44] D.L. Jutla and P. Bodorik, Sociotechnical architecture for online privacy, *IEEE Security & Privacy*, 2005, pp. 29–39.
[45] N. Li, J.C. Mitchell and W.H. Winsborough, Beyond proof-of-compliance: Security analysis in trust management, *Journal of the ACM* **52**(3) (2005), 474–514.
[46] R. Leenes, J. Schallaböck and M. Hansen, Prime white paper v2, 2007.
[47] A. Lysyanskaya, Signature schemes and applications to cryptographic protocol design, PhD thesis, Massachusetts Institute of Technology, 2002.
[48] L. Nguyen and R. Safavi-Naini, Dynamic k-times anonymous authentication, in: *ACNS*, J. Ioannidis, A.D. Keromytis and M. Yung, eds, Lecture Notes in Computer Science, vol. 3531, 2005, pp. 318–333.
[49] T. Olsen, T. Mahler, C. Seddon, V. Cooper, S. Williams, M. Valdes and S.M. Arias, Privacy & identity management, Technical report, Senter for rettsinformatikk, 2007.
[50] J. Ni, N. Li and W.H. Winsborough, Automated trust negotiation using cryptographic credentials, in: *Proc. of the 12th ACM Conference on Computer and Communications Security (CCS 2005)*, Alexandria, VA, USA, November 2005.
[51] Organization for Economic Co-operation and Development, OECD guidelines on the protection of privacy and transborder flows of personal data, 1980; http://www.oecd.org/document/18/0,2340,en_2649_34255_1815186_119820_1_1_1,00.html.
[52] A.S. Patrick and S. Kenny, From privacy legislation to interface design: Implementing information privacy in human-computer interfaces, in: *PET2003*, Dresden, 2003.
[53] PRIME Consortium, Architecture v3, Deliverable D14.2.c, 2008.
[54] PRIME Consortium, Framework v3, Deliverable D14.1.c, 2008.
[55] PRIME Consortium, Requirements for privacy enhancing tools (forthcoming), Deliverable, 2008.
[56] C.D. Raab, Perspectives on "personal identity", *BT Technology Journal* **23** (2005).
[57] J. Rachels, Why privacy is important, in: *Philosophy and Public Affairs*, 1975, pp. 323–333.
[58] R.L. Rivest, A. Shamir and L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, *Communications of the ACM* **21**(2) (1978), 120–126.
[59] T. Ryutov, L. Zhou, C. Neuman, T. Leithead and K.E. Seamons, Adaptive trust negotiation and access control, in: *Proc. of the 10th ACM Symposium on Access Control Models and Technologies*, Stockholm, Sweden, June 2005.

[60] K.E. Seamons, W. Winsborough and M. Winslett, Internet credential acceptance policies, in: *Proc. of the Workshop on Logic Programming for Internet Applications*, Leuven, Belgium, July 1997.

[61] K. Seamons, M. Winslett and T. Yu, Limiting the disclosure of access control policies during automated trust negotiation, in: *Proc. of the Network and Distributed System Security Symposium (NDSS 2001)*, San Diego, CA, USA, April 2001.

[62] A. Shostack, People won't pay for privacy, reconsidered, 2003.

[63] F. Stalder, The failure of privacy enhancing technologies (pets) and the voiding of privacy, *Sociological Research Online* **7**(2) (2002).

[64] I. Teranishi, J. Furukawa and K. Sako, k-times anonymous authentication (extended abstract), in: *ASIACRYPT*, P.J. Lee, ed., *Lecture Notes in Computer Science*, vol. 3329, Springer, 2004, pp. 308–322.

[65] Trusted Computing Group, TCG TPM specification version 1.2, url: www.trustedcomputinggroup.org.

[66] T.W. van der Horst, T. Sundelin, K.E. Seamons and C.D. Knutson, Mobile trust negotiation: Authentication and authorization in dynamic mobile networks, in: *Proc. of the Eighth IFIP Conference on Communications and Multimedia Security*, Lake Windermere, England, September 2004.

[67] Web services policy framework, March 2006; http://www.ibm.com/developerworks/webservices/library/specification/ws-polfram/?S_TACT=105AGX04&S_CMP=LP.

[68] A. Westin, *Privacy and Freedom*, Atheneum, New York, 1967.

[69] M. Winslett, N. Ching, V. Jones and I. Slepchin, Assuring security and privacy for digital library transactions on the web: Client and server security policies, in: *Proc. of the ADL'97 – Forum on Research and Tech. Advances in Digital Libraries*, Washington, DC, USA, May 1997.

[70] World Wide Web Consortium, *The Platform for Privacy Preferences 1.1 (P3P1.1) Specification*, July 2005; http://www.w3.org/TR/2005/WD-P3P11-20050701.

[71] World Wide Web Consortium, *A P3P Preference Exchange Language 1.0 (APPEL1.0)*, April 2002; http://www.w3.org/TR/P3P-preferences/.

[72] T. Yu, X. Ma and M. Winslett, An efficient complete strategy for automated trust negotiation over the internet, in: *Proc. of the 7th ACM Computer and Communication Security*, Athens, Greece, November 2000.

[73] T. Yu and M. Winslett, A unified scheme for resource protection in automated trust negotiation, in: *Proc. of the IEEE Symposium on Security and Privacy*, Berkeley, California, May 2003.

[74] T. Yu, M. Winslett and K.E. Seamons, Interoperable strategies in automated trust negotiation, in: *Proc. of the 8th ACM Conference on Computer and Communications Security (CCS 2001)*, Philadelphia, Pennsylvania, USA, November 2001.

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

[76] T. Zarsky, Mine your own business!: Making the case for the implications of the data mining or personal information in the forum of public opinion, *Yale Journal of Law & Technology* **5** (2002), 17–47.

[77] T. Zarsky, Desperately seeking solutions: Using implementation-based solutions for the troubles of information privacy in the age of data mining and the internet society, *Maine Law Review* **56** (2004), 14–59.