BlindIdM: A Privacy-Preserving Approach for Identity Management as a Service

David Nuñez · Isaac Agudo

Received: date / Accepted: date

Abstract Identity management is an almost indispensable component of today's organizations and companies, as it plays a key role in authentication and access control; however, at the same time it is widely recognized as a costly and time-consuming task. The advent of cloud computing technologies, together with the promise of flexible, cheap and efficient provision of services, has provided the opportunity to externalize such a common process, shaping what has been called *Identity Management as a Service* (*IDaaS*). Nevertheless, as in the case of other cloud-based services, IDaaS brings with it great concerns regarding security and privacy, such as the loss of control over the outsourced data. In this paper we analyze these concerns and propose BlindIdM, a model for privacy-preserving IDaaS with a focus on data privacy protection. In particular, we describe how a SAML-based system can be augmented to employ proxy re-encryption techniques for achieving data confidentiality with respect to the cloud provider, while preserving the ability to supply the identity service. This is an innovative contribution to both the privacy and identity management landscapes.

Keywords Identity management, Cloud computing, Proxy re-encryption, SAML, Privacy, Data confidentiality

D. Nuñez · I. Agudo
Network, Information and Computer Security Laboratory, Universidad de Málaga, Málaga, Spain
E-mail: dnunez@lcc.uma.es

I. Agudo
E-mail: isaac@lcc.uma.es

1 Introduction

Cloud computing has recently burst onto the technology and business scenes, promising great technical and economic advantages. One of the principal benefits of cloud computing is that it represents a model of utility computing, capable of offering on-demand provisioning of computing resources, such as storage, processing and networking. This provision of resources is metered for billing purposes, making a "pay-as-you-go" model possible that permits companies and organizations to transform capital expenditures, such as acquisition of specific hardware, into operational expenditures; this paradigm can be contrasted with previous models, based on the acquisition of equipment and software licences. The main benefits that organisations expect from adopting the cloud computing paradigm are an improved flexibility and scalability of their IT services, as well as the resulting cost savings from the outsourcing of such services [1].

Within the internal processes of most organizations, identity management stands out for its ubiquitous nature, as it plays a key role in authentication and access control. However, it also introduces an overhead in cost and time, and in most cases, specialized applications and personnel are required for setting up and integrating identity management systems, as well as for managing identity information. As has already happened for other kinds of services, the cloud paradigm represents an innovative opportunity to externalize the identity management processes, offering what has been called *Identity Management as a Service* (*IDaaS*) [2]. Identity Management as a Service is the cloud industry's response to the problem of identity management within companies and organizations, allowing them to outsource the identity management service from their internal infrastructures and deploy it in the cloud provider. In other words, it permits moving identity management from an *on-premise* delivery model to an *on-demand* model. Additionally, IDaaS opens up a new business opportunity for cloud providers and vendors, broadening their service offering.

The advent of cloud computing has raised great expectations regarding efficiency, cost reduction and simplification of business processes, but at the same time has also increased security and privacy risks. This very same conflict also applies to the IDaaS case: although it offers organizations a great opportunity to cut capital costs (as well as some operational ones, such as specialized personnel), it also introduces a variant of one of the classic problems of cloud computing: the loss of control over outsourced data, which in this case is information about users' identity. For instance, according to a recent survey from Cisco to IT specialists and decision makers [3], data protection is regarded as the top barrier that impedes the migration to the cloud.

Current identity providers take the role of stewards of user data, being responsible for storing and managing this information. That is, users entrust their personal information to identity providers, which then have a privileged position in order to gain information about their users: they are not only able to read users' data that is in their custody, but also to keep a registry of the access patterns of the users to service providers, making the construction of a model of their behaviour possible. Although there are several regulatory, ethical and economic reasons for discouraging this possibility, the fact is that nothing actually prevents identity providers from accessing users' information at will. As a report from Gartner about risk in cloud computing states: *"If your data can be read at your provider's site, then you have to assume that it will be read"* [4]. Even if we assume that the identity provider is not dishonest and that its internal policy is respectful regarding identity information, it is still possible that a privacy disclosure occurs, for example through security breaches, insider attacks, or legal requests [5][6].

Traditionally, cloud providers have tackled these problems defining Service Level Agreements (SLAs) and internal security policies; however, these measures simply reduce this issue to a trust problem. Nothing actually prevents providers from breaking these agreements and policies; users simply *trust* them not do it. In other words, there is an important trust problem, inherent to cloud computing – users want to have access to services but, at the same time, they are unwilling to provide their data to entities that they do not necessarily trust. For these reasons, it is desirable to have more advanced security mechanisms at our disposal that enable users to benefit from cloud computing and still preserve their privacy and the control over their information.

The principal motivation behind this work is putting the identity provider into the cloud landscape, where data storage and processing could be offered by possibly untrusted cloud providers, but still offer an identity management service that guarantees user's privacy and control. To this end, we define BlindIdM, a privacy-preserving IDaaS system where identity information is stored and processed in a blind manner, removing the necessity of trusting that the cloud identity provider will not read the data. Such a concept is a novel contribution to both the field of identity management and privacy-enhanced technologies. Our model, which uses SAML 2.0 as the underlying identity management protocol, applies proxy re-encryption techniques to achieve end-to-end confidentiality of the identity information, while allowing the cloud to provide an identity service.

The rest of this paper is organized as follows: In Section 2, we describe the path to Identity Management as a Service and provide an analysis of its associated risks and challenges; we also present our point of view regarding a distribution of the identity management competences through the cloud ecosystem. In Section 3, we explain our proposal in detail, including its motivations and scope, and give an overview on the underlying technologies: SAML and proxy re-encryption. In Section 4, we present some of the work related to identity and privacy management that is relevant to our proposal. Finally, Section 5 concludes the paper and future work is outlined.

2 The Path to Identity Management as a Service

Identity information is steadily becoming an essential enabler of today's digital society, as it is considered a key component in the interactions between end-users, service providers, and intermediaries. At the same time, it is also becoming more and more valuable for the organizations that manage this kind of information because of its usefulness for marketing and strategic development purposes or, simply, to be sold to interested third parties [7]. Thus, identity management remains an important challenge in the field of information security and privacy, and spans several subareas, such as usability and user experience, authentication methods, or trust and reputation management [8].

Within the organizations' environment, identity management is one of the most commonly deployed services because of its importance for authentication and access control. However, it is regarded by enterprises as one of the most time-consuming and complex tasks within their internal business processes. It introduces an overhead in cost and time, and in most cases, specific applications and personnel are required for managing, integrating and maintaining this service. This is even more troublesome when some kind of identity service is offered to external users, such as clients, contractors, or providers.

An identity management system (IMS) facilitates the creation, storage, and usage of the identity information of the individuals from a organization [9]. Traditionally, identity management systems were designed to be used internally in the organizations and companies, in a centralized and local manner, which has been called the *silo model*. However, as the Internet has gained in popularity, the number of possible interactions that a user can have with service and resource providers has increased dramatically. This fact leads to an unwanted effect called *identity fragmentation*, since users are then obliged to register several accounts, one for each service provider; that is, their identity information is partially replicated and fragmented throughout a group of service providers. Furthermore, each of these fragments of identity is normally associated with passwords that must be memorized by the users, which is prone to usability and security problems, such as password reuse. The problem of identity fragmentation evidences the drawbacks of the traditional isolated model of identity management, and has motivated the development of more flexible schemes that are centered on enhancing the interoperability.

2.1 Federated Identity Management

Federated Identity Management (*FIM*) is a solution to overcome these difficulties. FIM is a set of distributed technologies and processes that enable information portability between different domains, which permits both a dynamic distribution of identity information and delegation of associated tasks, such as authentication or user provisioning. Thus, organizations coordinate with each other to form federations for exchanging identity information. One of the key aspects of this model is the establishment of trust relationships between the members of the federation, which enables them to believe the statements made within the federation. This way, although users are authenticated by their local organization, they are able to access services and resources from other organizations of the federation. SAML [10], Shibboleth [11] and WS-Federation [12] are examples of systems and standards for federated identity management.

The parties involved in a federated identity interaction are required to mutually exchange identity information for identification and authentication purposes regardless of whether they have previous knowledge of each others' identity information or not. The main actors that participate in these interactions are [13],[14]:

- *Users*, the subjects of the identity information; most of the times they are also the principal source of this information. Users are generally the actors that request resources and services through their interaction with applications and online services. Users perform this interaction through a *user agent*, which is usually a browser, but it could also be a specific application.
- *Service Providers* (*SP*), the entities that provide services and resources to users or other entities. In a federated identity management context, service providers outsource the processes of authentication and management of users to identity providers. Because of this, service providers act as consumers of user's identity information, following a determined identity management protocol.
- *Identity Providers* (*IdP*), which are specialized entities that are able to authenticate users and to provide the result of this authentication to service providers, without revealing additional information about the user. The information that they exchange with service providers may even be just a statement about the success of the authentication of the user, enabling the user to access the service anonymously. Identity providers are also responsible for managing the identity information of their associated users, and in some cases, they may certify it.

Figure 1 shows a high-level view of a federated identity setting, where a host organization acts as a federated identity provider. In this setting, an employee from the host organization requests a service from the service provider, who in turn asks the organization for identity information about its employee.

Before accepting the supplied identity information, the service provider must trust the host organization, acknowledging it as a reliable identity provider. Trust in this case is normally achieved out of band through some physical transaction such as a legal agreement, and later reflected in the identity federation system through some technical mechanisms such as WS-Trust or SAML Metadata; in practice, each consumer entity has a list of trusted issuers of identity information.

2.2 Identity Management as a Service

The federated identity model is widely used in organizations, deployed as an on-premise service. Although

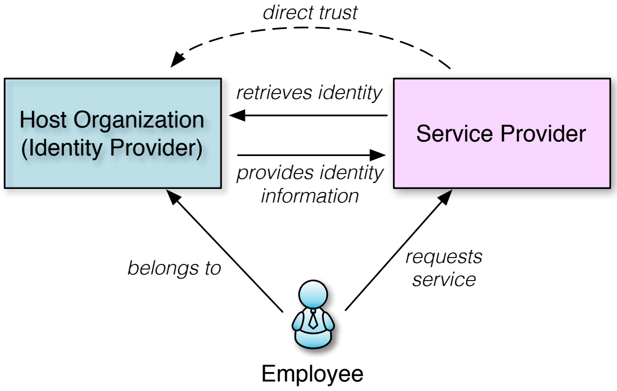

Fig. 1 Federated Identity Management System

it has led to great advantages with respect to interoperability of identities, it has also introduced cost and time overheads, since it usually requires specialized applications and personnel for setting up, integrating and managing this process.

However, the emergence of the cloud as a ubiquitous technology within today's organizations, has led to Identity Management as a Service, a natural answer from the cloud industry to the enterprise identity management problem. Examples of such cloud-based identity services are Windows Azure Active Directory [15] and CA CloudMinder Identity Management [16]. IDaaS can be seen as a refinement of the federated model, which takes the efficiency of the cloud in its favour for offering specialized outsourcing of identity management. Among the benefits of Identity Management as a Service we find:

- More flexibility, scalability and stability for high demand environments, with a growing number of users and thousands of identities.
- Reduction of costs, since IDaaS providers can focus on providing more efficient and specialized identity services to organizations.
- Better security measures and mechanisms, implemented in dedicated systems and facilities.
- Improved compliance and business processes audits due to the high specialization and security standards that an IDaaS provider can achieve. These providers can also implement common policies for all their customers at a lower cost.

2.3 Rethinking Identity Management as a Service

The adoption of the Identity Management as a Service approach by organizations for consuming and providing identity services constitutes a great opportunity for both the cloud computing and identity management industries. The cloud ecosystem offers the opportunity of defining a holistic and more flexible model for identity management that takes advantage of all of its characteristics. This model must be designed taking into account the varied security and privacy challenges that occur in the intersection of identity management and cloud coumputing.

A mere relocation of an identity provider into the cloud is not enough. Let us assume a scenario where a host organization outsources its whole identity management system to a cloud identity provider. In this case, the cloud identity provider is in full control of the identity management processes of the host organization, including authentication, storage of identity information, identity management protocol execution, etc. Although such a setting may be useful for certain profiles of companies and organizations, it requires a high level of trust in the cloud provider, since it implies a lot of control in the hands of an external entity.

Taking into account the current technological and corporate context, where the cloud has gained great relevance due to its numerous advantages, the existence of a sole entity acting as identity provider will not take advantage of all the possibilities provided by the cloud. What is more, this entity monopolizes the roles, services and processes involved in the management of identity information, increasing the associated risks; the inherent flexibility of the cloud is therefore lost. It is necessary to decompose these roles, services and processes and to distribute and orchestrate them throughout the ecosystem of the cloud to achieve an appropriate IDaaS. This allows us to achieve fine-grained combination and customization of the needs, risks and responsibilities of all the parties involved in identity interactions.

We advocate for a model for the distribution of identity management competences throughout the cloud ecosystem that permits:

- Optimizing the use of resources, including cloud providers and in-house infrastructures.
- Enhancing privacy and confidentiality of identity information and reducing security risks.
- Increasing the control over the identity management processes.
- Opening up new business opportunities for cloud providers.

If we break down an identity management system from a more functional perspective, we can extract a set of *competences* or functional responsibilities that describes in more detail the different roles and functions that take place in an identity management system. We identify the following competences:

Identity management protocol. This competence strictly refers to the fulfillment of the underlying identity management protocol. It is also distributed into the different entities participating in the interactions of the identity management system (users, identity providers, service providers, etc.). Federated identity services, which provide means for making authentication and authorization information compatible and usable between separate security domains, are part of this competence. For example, in the case of a SAML-enabled identity provider, this role is in charge of generating SAML responses after receiving a SAML request. This competence may be replicated within an identity provider in order to offer a multiprotocol gateway or identity bridges.

Authentication. This competence is in charge of verifying that the claims made by a user about his identity are true; in other words, to ensure that the user is who he says he is. This is normally achieved using some sort of credentials, such as passwords or X.509 certificates, that are linked to the corresponding user profile. Provision of strong and multi-factor authentication mechanisms is also a duty of this competence.

Storage of identity information. Identity information must be stored and managed using some kind of repository. Note that the entity in charge of this might not be the same as the one that provides the interface of the identity service. For example, an identity provider may externalize the storage of identity information to a third-party storage cloud provider, so the IDaaS model must be flexible enough to support this possibility. This competence includes the functionality of directory services, such as LDAP or Active Directory.

Protection of data. Identity information is usually protected only during the the communications phase; the protection of data *in transit* is a widely studied problem with multiple solutions, such as TLS/SSL at transport level, and WS-Security at application level. Identity management protocols rely on these solutions to provide security from a communication standpoint; for example, WS-Federation relies on the end-to-end security provided by WS-Security to convey identity information through SOAP messages.

However, the protection of data *at rest* is not considered in any identity management system, although it is one of the weakest points in the security chain; usually, data at rest is kept unprotected and, therefore, is left open to insider attacks or breaches due to security failures [17]. Furthermore, certain regulations, such as HIPAA and PCI, mandate that personal identifiable information must be protected using appropriated encryption techniques.

A trivial, but inefficient, solution to this problem could be to encrypt the data using a symmetric encryption algorithm, such as AES. Although it protects the data at rest, this option leads to a key management problem, since the user would have to share his secret key with the service providers, which implies difficulties for revocation and managing shared keys. This solution is the one used by most of the current outsourced storage providers, such as Amazon S3, since user's data is not intended to be shared with third parties; in this case, data is protected from external threats but is vulnerable to insider attacks.

Authoritative source. This competence is related to a source of identity information that is trusted in a certain context. Think of the Human Resources department of a company as an example; they have all the information regarding the employees of the company, this is why they become the authoritative source for information regarding the employees. In-house identity providers are usually connected to human resources databases to provide this information to other services. Authoritative sources also deal with the validity of the data, ensuring aspects such as correctness and freshness. In an identity management system, this competence may be either centralized or distributed. Usually, corporate directory services also act as the authoritative sources for identity information.

Certification. This competence is dedicated to providing assurance of the truthfulness of the identity information, in the form of a mechanism that provides some sort of evidence that permits verifying the validity of the information. For example, identity information may be digitally signed in order to ensure its veracity; in this case, the digital signature and the public key infrastructure are the mechanisms that implement the assurance. Usually the identity provider gives certification regarding the link of identity information to users of the system, primarily when using external services. This role is in charge of providing some evidence that the users accessing the services really do own the corresponding attributes.

Authorization and privilege management. This competence is devoted to the definition of user's entitlements, usually, through the specification of users' roles within the organization. This way, organizations can establish mappings between identities, permissions, actions and resources.

2.4 Risks and Challenges of Identity Management as a Service

There are multiple risks associated with Identity Management as a Service; most of them are a consequence of the identity providers managing and storing a large amount of identity information [18], while some of them are inherent to the cloud computing paradigm [17][5][19]. We identify the following principal risks:

- Identity providers are appealing targets to attackers as they represent a single point of failure because they centralize users' personal information; security breaches and insider attacks are potentially dangerous as they may disclose the personal information of a large number of users. The fact that this kind of information is protected by specific regulations, such as the European Data Protection Directive, in the case of the EU [20], demands a strong protection of its storage, processing and communication.
- Cloud providers are susceptible to being subpoenaed for users' data, in the case there is some legal, administrative or criminal investigation running [21]. What is worse, it is possible that providers respond positively to these requests for information, even if they are not made with the proper judicial guarantees, due to a lack of legal understanding.
- Identity providers are in a privileged position to collect additional information about users without their consent, such as the sites the user visits, for profiling purposes.
- In the absence of cryptographic means, it is not possible to actually limit the access of cloud providers to the data they must steward. That is, there is almost no risk of being discovered accessing users' information without their consent.
- Another major risk is the existence of cloud providers in foreign countries (i.e., located in a different country to the owner of the data) with different, and possibly conflicting, laws and regulations regarding privacy and data protection. For example, in the case of the US, the USA PATRIOT Act [22] allows the government to check the data that is processed or stored within its jurisdiction, even without the knowledge of the owner of the data.
- The security guarantees and requirements of the cloud providers are disparate. These requirements not only include technological aspects, but also policies regarding the hiring of staff, access to premises and equipment, physical security measures, etc.

Hence, it is obvious that externalizing the management of identity information to the cloud implies a loss of control for users and organizations. This in turn signifies an empowerment of cloud identity providers; that is, there is an inversion of the control over the identity information. This leads to the identity provider accumulating enough power for the users to incur damages, losses or risks in the case of a disclosure of private data.

3 BlindIdM: Privacy-Preserving IDaaS

In this section, we will describe our proposed system, BlindIdM, for realizing a privacy-preserving IDaaS in the cloud. Our proposal complies with our vision of Identity Management as a Service, where competences are distributed throughout the entities of the cloud ecosystem. In contrast to a full outsourcing of the identity management system, we have opted for a hybrid approach, where the authentication remains on-premises at the host organization. The novel aspect of our proposal lies in the protection of data: the host organization encrypts users' identity information prior to outsourcing it to the cloud, in such a way that it is still usable by the cloud identity provider without being able to be read.

We will firstly describe the motivation behind this proposal and then we will discuss some of the potential benefits and advantages that may stem from its adoption; secondly, we will specify the context of our approach by establishing its scope and specifying the trust model that we will consider; thirdly, we will briefly describe the underlying technologies used in our solution: SAML for the underlying identity management framework and proxy re-encryption to implement the cryptographic protection; then, we will explain in detail our proposal for privacy-preserving Identity Management as a Service; and finally, we will provide an analysis of our proposal.

3.1 Motivation and Incentives

In an IDaaS scenario, organizations entrust their corporate identity information to cloud identity providers, which are then responsible for storing and managing this information. These systems rely on the existence of a strong relationship of trust between the organizations and the cloud identity providers, since they trust that their identity information will be managed properly and that the provider will respect the confidentiality; however, current cloud providers do not implement real mechanisms for preventing themselves from betraying this trust. This concern led us to conceive of the concept of *Blind Identity Management* (BlindIdM), a system

whereby the cloud identity provider is able to offer an identity information service, without knowing the actual information of the users; that is, it provides this service in a *blind*[1] manner.

[1] The term *blind* is used here in an analogous way as in *blind* signature, which is a signature scheme that enables the signer to perform a signature without knowing the content of the underlying message.

This is a great innovation with respect to current identity management systems, where users' identity information is managed by the identity provider and the user is obliged to trust that the provider will make proper use of his data and will guarantee its protection. Our intention is that this model will enable organizations to choose a cloud identity provider without necessarily establishing a strong bond of trust with it; i.e., they do not have to trust that the cloud identity provider will respect data privacy. Instead, the sturdiness of the underlying cryptographic schemes should be sufficient to guarantee such protection.

It is interesting to think about what kind of incentives may motivate a cloud identity provider to offer its services in a blind manner. From a strictly economic point of view, it may not make sense to still provide these services for free, since they will probably incur more expenses as a result of implementing additional security mechanisms. Furthermore, they will lose control over the user's data, which is currently a valuable asset. Still, there are some incentives that could encourage cloud identity providers to offer such a blind service.

3.1.1 Compliance with Data Privacy Laws and Regulations

In an IDaaS setting, one of the consequences for cloud identity providers is that they can be seen in the eyes of the law as stores and processors of *Personal Identifiable Information*(*PII*). As a consequence, they are obliged to comply with specific laws and regulations regarding data protection, such as the EU Data Protection Directive and the Health Insurance Portability and Accountability Act (HIPPA)[2] [23]. Some of these regulations demand that personal identifiable information must be protected using appropriated encryption techniques.

Therefore, a privacy-preserving approach like ours, which achieves data confidentiality through encryption mechanisms and permits cloud identity providers to offer identity services without having the chance to access the data, could be very useful to help them to comply with these kinds of regulations. We argue that, given the proper cryptographic safeguards, encrypted data is not private anymore, and could even be freely distributed without compromising users' privacy; however, we note that currently there are some legal uncertainties regarding this matter [24].

3.1.2 Minimization of Liability

Currently there is a lot of discussion, especially from the cloud industry and lawmakers, with regard to liability in cloud computing due to its nature of outsourced service provision. Although cloud providers currently try to reduce their liability through specific clauses in SLAs, legal responsibility for the data in the cloud also lies on the side of the cloud provider. There are a lot of examples from blog sites, Internet forums, or file hosting services (such as the Megaupload case in 2012 [25]), where the owners of these services are indicted for hosting illegal or defamatory material, even though they have not generated said content.

In contrast, given that in our proposal outsourced data is encrypted prior to arriving the cloud and the cloud provider does not hold the decryption keys, liability is drastically minimized, as they are unable to read user's data. Take a shipping service as an analogy: they will not be liable for any illegal or dangerous item delivered through their service, since they cannot open the packages and inspect their content (or at least, every delivered package). As a consequence of this, users should be the ones designated as liable and subject to the enforcement of key disclosure laws.

3.1.3 Data Confidentiality as an Added Value

An interesting incentive for cloud providers could be the possibility of offering secure data processing and confidentiality as an added value. Setting aside legal and regulatory aspects, this model could help a cloud provider to offer a competitive advantage over the rest. We foresee the model of Blind Identity Management as a technical starting point for a business model based on the respect for users' privacy and data confidentiality. Currently, there are some cloud services, such as PrivateSky [26] or CipherCloud [27] that have built their business model on data confidentiality as an added value.

3.2 Trust Model and Assumptions

In our model, we will assume a federated identity setting, similar to that shown in Figure 1, but where the host organization partially outsources the identity management processes to a cloud identity provider, while retaining the authentication service on-premises. The cloud identity provider now acts as an intermediary in the identity interactions, and is also in charge for storing and supplying identity information; Figure 2 shows this setting. Optionally, other kinds of actors may come

[2] Note that HIPAA is focused on the healthcare sector.

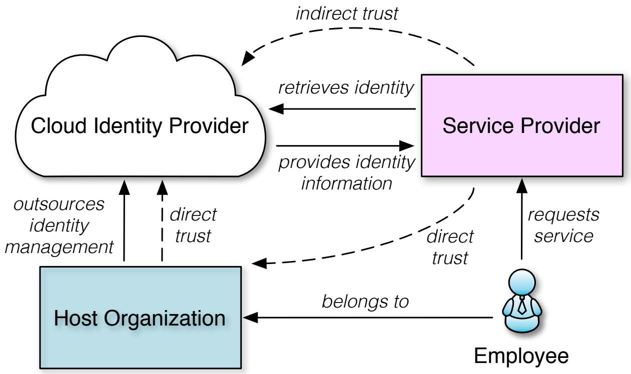

Fig. 2 Relationships between entities

into play such as attribute issuing authorities, certification providers, identity brokers, etc; however, we will restrict the scope of this work to the basic case.

The goal of our approach is to provide a means for constructing *blind identity providers*, which could be capable of operating without having access to users' information. In other words, we consider the cloud identity provider as an adversary. Based on the definitions given in [28], we identify three types of cloud identity providers depending on their capabilities and their level of trust:

- Trusted: The identity provider is a fully trusted entity, which provides an identity service correctly and truthfully. For the sake of convenience, this is the type of provider which is generally assumed, since users entrust their personal information to providers without demanding strong protection mechanisms and, hence, supposing that the providers will always be trustworthy. However, as we have shown before, this is not always the case, so it is not a realistic model of identity provider.
- Honest-but-curious: The identity provider follows the agreed protocol correctly, but it also stores or collects information about the users without their consent. Depending of the nature of this information, there are two distinct, but not exclusive, subtypes of honest-but-curious identity providers:
  - Data-curious: The identity provider has some incentive to read users' data that is in its custody. Such an incentive could be, for example, selling users' private information to third-parties for advertising or fraudulent operations; other motivations could be industrial espionage or political repression.
  - Access-curious: The provider collects information related to the access patterns of the user, which enables it to track users' behavior and threaten their privacy.

  We also assume that honest-but-curious identity providers do not collude with service providers in order to read users' data; in fact, once a service provider has access to an attribute of the user, it would be trivial to share it with the identity provider.
- Malicious: The provider has the possibility to actively deceive the user, read user's data and collect access patterns; it may also collude with other entities, such as service providers, to do so. A provider of this type may not follow the agreed protocols, so users cannot trust that they will do so. This is the most difficult type of adversary; however, it is the most realistic, since it actually models all the capabilities of a real identity provider.

With this classification in mind, we will restrict the work in this paper to a *data-curious provider*, which behaves correctly with respect to protocol fulfillment, but has no hindrance to try to access users' data. We will assume then that the identity provider may have some incentive to read users' data without their consent, but will not try to track users' behaviour and access patterns.

The problem that arises from considering access-curious providers has been widely studied [29], as it is what one normally encounters when privacy is addressed in the context of identity management; anonymization techniques, such as pseudonyms [30], are among the solutions that are usually proposed in this respect. As aforementioned, in this paper we will not tackle this problem and we will focus instead on protecting data privacy; however, a more complete solution that takes this issue into account is left open as future work.

As stated before, Figure 2 depicts the main interactions in our proposed model. With regard to trust relationships, the introduction of the cloud identity provider makes them more complex than in the federated identity setting. On the one hand, we still assume that the service provider fully trusts the host organization as a reliable and valid source of identity information; this trust is achieved as in the federated case, through out-of-band agreements and metadata. On the other hand, since the host organization outsources part or all of its identity management system, it is clear that the organization must have some level of trust in the cloud identity provider. In this case, trust is reflected in SLAs and in metadata as well. As a consequence of these direct trust relationships, we assume that in this setting the service provider indirectly trusts the cloud identity provider, as there is an implicit chain of trust between the two entities. That is, there is no explicit agreement or metadata that expresses this trust relationship, but the service provider can be confident of the trustworthiness of the cloud provider.

3.3 Underlying Technologies

Our system is based in two main technologies: SAML 2.0 as the underlying identity management protocol, and proxy re-encryption as the cryptographic protection. We have chosen SAML because of its wide adoption, its extensibility and its ingrained mechanisms for establishing trust between the entities, and proxy re-encryption because it enables to ensure users' data confidentiality with respect to the cloud provider while preserving the ability to supply the identity service. In the next subsections, we describe in more detail both technologies.

3.3.1 SAML 2.0

SAML 2.0 (Security Assertion Markup Language) [10] is a standard XML-based framework that enables the description and exchange of identity information between different security domains. With SAML, identity information is expressed in the form of assertions, which are a set of statements about a subject; these statements cover different aspects, such as authentication, authorization and identity attributes.

The SAML framework also specifies the protocols for issuing and exchanging assertions, such as the Authentication Request protocol, a request/response protocol that permits entities to ask for an authentication statement, and optionally identity attributes. In this protocol, the requester sends a SAML `AuthnRequest` message to an identity provider, which in turn replies with a SAML `Response` containing assertions about the request. The technical details about how to achieve this message exchange depend on the specific SAML bindings and profiles in use; in this paper, we will use the Web Browser SSO Profile and HTTP POST Binding as a basis for the identity interactions. The Authentication Request protocol also permits the hybrid approach for authentication we have chosen, as this possibility is considered in the SAML specification. In this case, the cloud identity provider acts as a *proxying identity provider*, and the host organization is the *authentication provider.*

SAML attributes are used to express identity information about the subject of the assertion; Figure 3 shows an example of such element. In our proposal, we make extensive use of this construction, as we take it as the basic medium for conveying encrypted identity information.

```
<saml:Attribute Name="givenName"
    ext:OriginalIssuer="http://idp.host.org"
    xmlns:ext="urn:oasis:names:tc:SAML:attribute:ext">
    <saml:AttributeValue>John</saml:AttributeValue>
</saml:Attribute>
```

Fig. 3 SAML Attribute

In addition, SAML permits the expression of metadata for both service providers and identity providers using the SAML Metadata specification [31]. Metadata is what enables the expression of prior trust relationships and makes secure transactions possible.

3.3.2 Proxy Re-Encryption

With regard to the cryptographic protection, we have used proxy re-encryption techniques for allowing the cloud identity provider to share the identity information in a blind manner; in particular, we have used the scheme proposed by Ateniese, Fu, Green and Hohenberger in [32].

From a high-level viewpoint, a proxy re-encryption scheme is an asymmetric encryption scheme that permits a proxy to transform ciphertexts under Alice's public key into ciphertexts under Bob's public key, as shown in Figure 4. In order to do this, the proxy is given a re-encryption key $r_{A \rightarrow B}$, which makes this process possible.

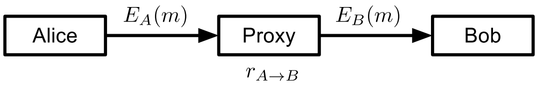

Fig. 4 General proxy re-encryption sequence

The notion of proxy re-encryption was introduced in 1998 by Blaze *et al.* [33]; their proposal, which is usually referred to as the BBS scheme, is bidirectional (it is trivial to obtain $r_{B \rightarrow A}$ from $r_{A \rightarrow B}$) and multihop (the re-encryption process is transitive), but not resistant to collusions.

Ateniese, Fu, Green and Hohenberger proposed in [32] new proxy re-encryption schemes based on bilinear pairings. In particular, they provided an initial basic scheme, which is subsequently extended throughout the paper to support additional functionalities. Their scheme is unidirectional, unihop and resistant to collusions. We chose this particular scheme for our implementation because of its simplicity and efficiency; it is explained in more detail below.

Green and Ateniese propose an identity-based proxy re-encryption scheme in [34]; however, this scheme is not resistant to collusions. An improved proposal that is secure against chosen-ciphertext attacks (CCA) is presented in [35], but again, it is not collusion-resistant.

In [36], Canetti and Hohenberger present a CCA-secure bidirectional scheme; based on this security model, Libert and Vergnaud propose in [37] a unidirectional scheme with chosen-ciphertext security in the standard model.

Another interesting proposal is presented in [38], where the authors define the notion of *key privacy* in the context of proxy re-encryption, which prevents the proxy to derive the identities of both sender and receiver from a re-encryption key.

AFGH scheme. As aforementioned, Ateniese *et al.* define in [32] a unidirectional, unihop and collusion-resistant proxy re-encryption scheme. This scheme is based in the ElGamal cryptosystem. Let $\mathbb{G}_1$ and $\mathbb{G}_2$ be two groups of prime order q, with a bilinear map $e : \mathbb{G}_1 \times \mathbb{G}_1 \rightarrow \mathbb{G}_2$; the global parameters are $g \in \mathbb{G}_1$ and $Z = e(g, g) \in \mathbb{G}_2$.

- *Key Generation* (KG): Alice selects a random integer $a \in \mathbb{Z}_q$ and generates her pair of secret and public keys, $s_A = a$ and $p_A = g^a$.
- *Re-Encryption Key Generation* (RKG): Alice takes Bob's public key p_B, and together with her secret key s_A, she computes the re-encryption key:

$$r_{A\rightarrow B} = (p_B)^{1/s_A} = g^{b/a} \in \mathbb{G}_1$$

- *First-level Encryption* (E_1): Anyone is able to encrypt messages intended only for Alice using her public key p_A. To encrypt a message $m \in \mathbb{G}_2$, one selects a random integer $k \in \mathbb{Z}_q$, and computes:

$$c_A = (e(p_A, g^k), mZ^k) = (Z^{ak}, mZ^k) \in \mathbb{G}_2 \times \mathbb{G}_2$$

- *Second-level Encryption* (E_2): To create second-level ciphertexts, which are re-encryptable, one computes:

$$c_A = (p_A^k, mZ^k) = (g^{ak}, mZ^k) \in \mathbb{G}_1 \times \mathbb{G}_2$$

- *Re-Encryption* (R): Anyone in possession of the re-encryption key $r_{A\rightarrow B}$ can transform a second-level ciphertext for Alice, $c_A = (\alpha, \beta)$, into a first-level ciphertext for Bob, by computing:

$$c_B = (e(\alpha, r_{A\rightarrow B}), \beta) = (Z^{bk}, mZ^k) \in \mathbb{G}_2 \times \mathbb{G}_2$$

- *First-level Decryption* (D_1): As in other asymmetric encryption schemes, Alice uses her secret key s_A to transform a ciphertext $c_A = (\alpha, \beta) \in \mathbb{G}_2 \times \mathbb{G}_2$ into the original message m. In order to do so, Alice computes:

$$m = \frac{\beta}{\alpha^{1/s_A}} = \frac{\beta}{\alpha^{1/a}}$$

$$\begin{tikzcd}
c_1 \in \mathbb{G}_2 \times \mathbb{G}_2 \arrow[dr, bend left, "D_1"] & & c_2 \in \mathbb{G}_1 \times \mathbb{G}_2 \arrow[ll, bend right, "R"'] \arrow[dl, bend right, "D_2"'] \\
& m \in \mathbb{G}_2 \arrow[ul, bend left, "E_1"] \arrow[ur, bend right, "E_2"'] &
\end{tikzcd}$$

Fig. 5 Transformations between plaintext and ciphertext spaces

- *Second-level Decryption* (D_2): For decrypting a ciphertext $c_A = (\alpha, \beta) \in \mathbb{G}_1 \times \mathbb{G}_2$, Alice computes:

$$m = \frac{\beta}{e(\alpha, g)^{1/s_A}} = \frac{\beta}{e(\alpha, g)^{1/a}}$$

This scheme uses two different ciphertext spaces; first-level ciphertexts are intended for non-delegatable messages, whereas second-level ciphertexts can be transformed into first-level ones through re-encryption. Figure 5 shows the different transformations defined in this scheme. It is important to note that our system will only use the second-level encryption function E_2 to encrypt and the first-level decryption function D_1 to decrypt, since our intention is to create re-encryptable ciphertexts with this scheme.

The AFGH scheme has the following properties:

- *Unidirectional*: The re-encryption key $r_{A\rightarrow B}$ cannot be used to derive the reverse one $r_{B\rightarrow A}$. This property is useful in settings where the trust relationship betwen Alice and Bob is not symmetrical.
- *Resistant to collusions*: If the proxy and Bob collude, they are not able to extract the secret key of Alice; at most, they can compute the weak secret $g^{1/a}$, but this information does not represent any gain to them.
- *Unihop*: This scheme is not transitive with respect to re-encryption; as shown in Figure 5, the re-encryption process transforms a ciphertext from one space to another, so this process cannot be repeated.

3.4 Description of BlindIdM

We now describe BlindIdM, a privacy-preserving model for blind Identity Management as a Service. In this model, as in the usual identity management systems, there are three main types of actors, namely, users, service providers and identity providers. In our scenario, the host organization (including all the employees) acts as the user, and the identity management of the organization is outsourced to a cloud identity provider.

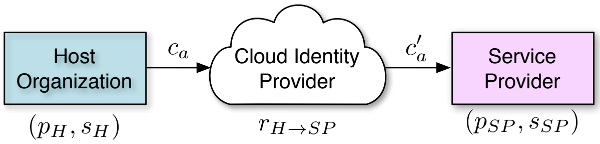

Fig. 6 Data flow of BlindIdM

These entities are capable of interacting following a predefined identity management protocol. From a high-level viewpoint, the goal of these interactions is the exchange of identity information, that generally flows from the user (in our case, from the host organization), acting as a source of information, to the service provider, acting as a consumer of information. BlindIdM permits this information to leave the source and arrive at its destination in an encrypted form, achieving end-to-end confidentiality. Our goal now is to describe how encrypted information can flow from the host organization to service providers, without the identity provider being able to read it.

A high-level diagram of our proposal is shown in Figure 6; this diagram depicts the main flow of information in our system, where the host organization encrypts the identity information under its public key p_H and sends it to the cloud identity provider. The use of proxy re-encryption enables the identity provider to transform these ciphertexts into encrypted attributes under the public key of the service provider, p_{SP}; in order to do so, the identity provider needs a re-encryption key $r_{H \to SP}$ generated by the host organization and provided beforehand.

We will now proceed to detail the steps of the operation of the BlindIdM system. Note that we are using SAML as the underlying protocol; here, we will describe an identity interaction using the SAML Authentication Request protocol. As stated before, we will use the AFGH scheme for proxy re-encryption explained in Section 3.3.2, where we describe in detail the cryptographic procedures for key generation, encryption, re-encryption and decryption. Since all these computations must be performed under the same set of system parameters $(\mathbb{G}_1, \mathbb{G}_2, e, g, Z)$, we will assume that these parameters are global and known by every party.

Phase 1. Generation of public and private keys. Both the host organization and the service provider create their pairs of public and private keys, respectively (p_H, s_H) and (p_{SP}, s_{SP}). For illustration purposes we will assume that there is only one service provider, but there could be any number of service providers. Furthermore, the service provider can create its pair of keys at any moment, as long as it is done before phase 3.

Phase 2. Encryption of identity information and outsourcing. The host organization must encrypt the identity information of its employees prior to externalizing it to the cloud. To do so, they use their public key p_H; for simplicity we will assume that identity information is in the form of attributes, where each attribute a is a tuple $(a.metadata, a.value)$, where $metadata$ describes any metadata about the attribute, including its name and format. Therefore, the identity information of each employee U is a pair $(ID_U, \{a : a \text{ is an attribute of the employee } U\})$, where ID_U is the identifier of such employee.

In our approach we encrypt just the attribute value, leaving the attribute metadata in clear, which eases the integration of our solution with existing directory services. The cloud provider is aware of the name of the attributes, but not of their content. It is also worth mentioning that we do not directly encrypt attribute values with the AFGH encryption function E_2; instead, we use a hybrid approach, encrypting a fresh key K_a for each attribute a, which is then passed to a symmetric encryption algorithm E_S (such as AES) for encrypting the attribute value. This way, the AFGH encryption primitive E_2 is only used to cipher a fixed-length input (the key K_a), whilst the symmetric algorithm performs the bulk of the work. This approach is used not only for efficiency, but for input length reasons, since attribute values have a wide range of possible lengths. Thus, an outsourced attribute c_a is generated in the following way:

$$c_a = (a.metadata, E_2(p_H, K_a), E_S(K_a, a.value))$$

An example of the SAML representation of an outsourced attribute c_a with an encrypted value is shown in Figure 7; this is the encrypted version of the same attribute shown in Figure 3. SAML permits putting any arbitrary content within the `AttributeValue` element, so we have used the XML Encryption specification [39] (used also in the SAML core specification) to express the hybrid encryption mechanisms and convey the cipher data and the key material.

Once the encryption process is complete, the host organization outsources the identity information to the cloud identity provider. The identity information in the cloud for each employee is in the form $(ID_U, \{c_a : c_a \text{ is an outsourced attribute of the employee } U\})$.

It is important to note that this same approach can be used to update attributes, so it does not represent any difficulty for our system; the new encrypted attribute simply substitutes the previous one.

```
<saml:Attribute Name="givenName"
    ext:OriginalIssuer="http://idp.host.org"
    xmlns:ext="urn:oasis:names:tc:SAML:attribute:ext">
    <saml:AttributeValue>
        <!-- Encrypted AttributeValue content -->
        <xenc:EncryptedData
            Type="http://www.w3.org/2001/04/xmlenc#Content">
            <!-- Symmetric encryption algorithm (AES-128) used
            -->
            <xenc:EncryptionMethod
                Algorithm=".../xmlenc#aes128-cbc"/>
            <!-- Encapsulated symmetric key using AFGH scheme -->
            <ds:KeyInfo>
                <xenc:EncryptedKey
                    Recipient="http://serviceprovider.com">
                    <!-- Proxy re-encryption algorithm (AFGH05)
                        used for key encapsulation -->
                    <xenc:EncryptionMethod
                        Algorithm="urn:proxyreencryption:afgh05"/>
                    <!-- Encapsulated symmetric key -->
                    <xenc:CipherData>
                        <xenc:CipherValue>
                            Ke41X+w...
                        </xenc:CipherValue>
                    </xenc:CipherData>
                </xenc:EncryptedKey>
            </ds:KeyInfo>
            <!-- Encrypted SAML Attribute Value content -->
            <xenc:CipherData>
                <xenc:CipherValue>39UCe3/sA...</xenc:CipherValue>
            </xenc:CipherData>
        </xenc:EncryptedData>
    </saml:AttributeValue>
</saml:Attribute>
```

Fig. 7 SAML Attribute with encrypted AttributeValue content

Phase 3. Trust establishment and generation of re-encryption keys. During this phase, service providers establish a trust relationship with the host organization, which is needed for deeming as valid the claims it makes. This relationship is bidirectional, since the host organization must also trust the service provider in order to release the identity information. As in the case of federated identity management, this trust relationship is usually a consequence of a prior out-of-band agreement.

SAML permits the expression of metadata for both service providers and identity providers using the SAML Metadata specification [31]; indeed, metadata is essential to the proper operation of some of the SAML protocols. The publication of keys and certificates, such as X.509 certificates, through the `KeyDescriptor` element, is among the crucial aspects that are covered in the metadata; we can make use of this method to publish the service provider's public key p_{SP}. SAML also permits expressing which attributes are required during authentication requests, using a specific element called `AttributeConsumingService`. Figure 8 shows an extract of the metadata file of the service provider, where these elements appear.

The cloud identity provider also needs a metadata file, but in this case, it does not require special attention, as the cloud provider does not have any key material prone to be distributed, other than its X.509 certificates. This metadata also contains the information about service endpoints for SAML protocols.

```
<md:EntityDescriptor
    entityID="http://www.serviceprovider.com" ...>
    <md:SPSSODescriptor ...>
        <!-- Proxy re-encryption public key of SP -->
        <md:KeyDescriptor use="encryption">
            <ds:KeyInfo>
                <ds:KeyName>
                    Proxy Re-Encryption Public Key
                </ds:KeyName>
                <ds:KeyValue>
                    <pre:PublicKey
                        Type="urn:proxyreencryption:afgh05">
                        YTdUs3...
                    </pre:PublicKey>
                </ds:KeyValue>
            </ds:KeyInfo>
        </md:KeyDescriptor>
        ...
        <!-- AttributeConsumingService elements -->
        <md:AttributeConsumingService index="1">
            <md:ServiceName>Service Provider</md:ServiceName>
            <md:RequestedAttribute
                Name="givenName" isRequired="false"/>
            <md:RequestedAttribute
                Name="cn" isRequired="true"/>
        </md:AttributeConsumingService>
        ...
    </md:SPSSODescriptor>
</md:EntityDescriptor>
```

Fig. 8 SAML Metadata of the service provider

Once the host organization trust a certain SP, they use its public key p_{SP} together with their private key s_H to create the re-encryption key $r_{H \to SP}$, which is then sent to the cloud identity provider. The host organization obtains the public key from the service provider's metadata. This key allows the identity provider to re-encrypt the ciphertexts in order to be decryptable by the service provider using its private key s_{SP}. The re-encryption key can be seen also as an authorization token, and can be revoked by the host organization by asking the cloud identity provider to remove it. Bear in mind that we are assuming an honest-but-curious cloud provider, which will follow the instructions given by the host organization. Another possibility could be to encrypt again the attributes with a new public key p'_H, and upload them to the cloud provider; this option is highly inefficient but does not require any other changes in our model.

Phase 4. Identity information interaction. Once our system is properly deployed, an employee may want to retrieve a resource from the service provider, which requires authentication and additional identity information from the employee. The goal in this phase is the dispatch of identity information of an employee, which is stored in the cloud identity provider, to the service provider. Moreover, the authentication takes place within the hosted organization, so its result must also be communicated to the service provider. As stated be-

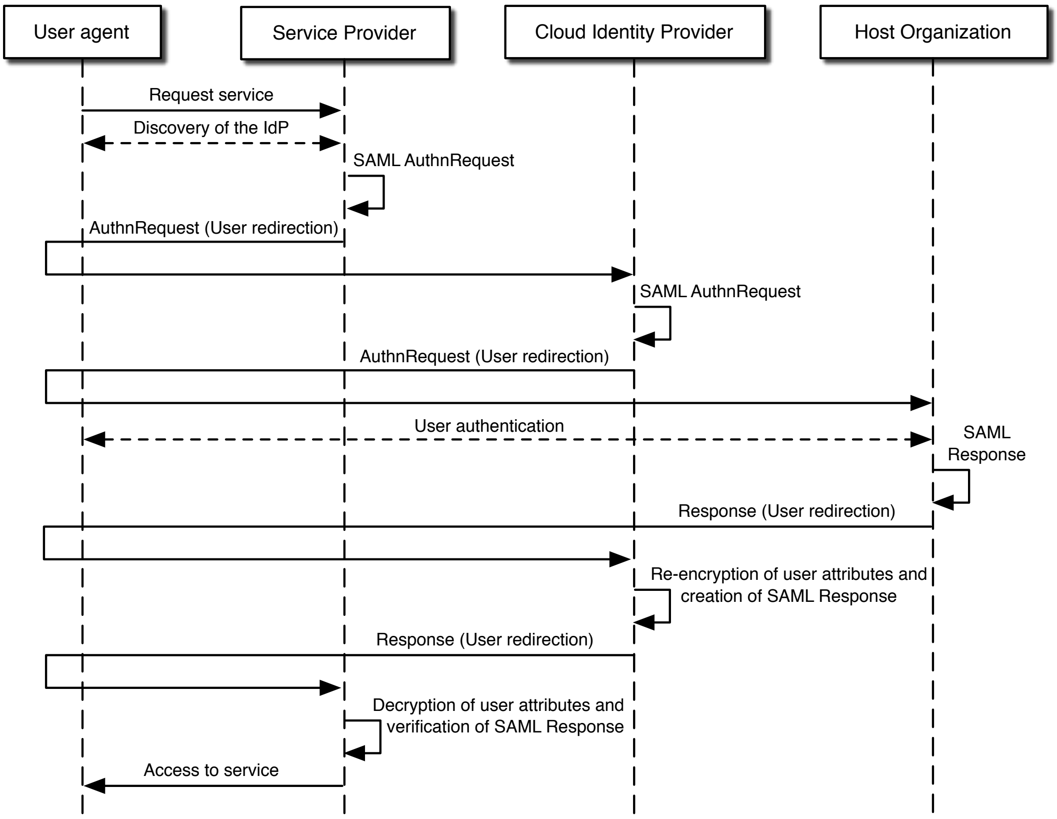
User agent
Service Provider
Cloud Identity Provider
Host Organization
Request service
Discovery of the IdP
SAML AuthnRequest
AuthnRequest (User redirection)
SAML AuthnRequest
AuthnRequest (User redirection)
User authentication
SAML Response
Response (User redirection)
Re-encryption of user attributes and creation of SAML Response
Response (User redirection)
Decryption of user attributes and verification of SAML Response
Access to service

Fig. 9 Sequence diagram of the authentication request

```
<saml:Assertion ID="#ASSERTION_ID" IssueInstant="2012-05-06T11:39:08Z" Version="2.0">
    <saml:Issuer>https://cloudidp.com</saml:Issuer>
    <ds:Signature xmlns:ds="http://www.w3.org/2000/09/xmldsig#">...</ds:Signature>
    <saml:Subject>
        <saml:NameID Format="urn:oasis:names:tc:SAML:1.1:nameid-format:emailAddress">
            jdoe@host.org
        </saml:NameID>
        <saml:SubjectConfirmation Method="urn:oasis:names:tc:SAML:2.0:cm:bearer">
            <saml:SubjectConfirmationData Recipient="http://serviceprovider.com"
                InResponseTo="#REQUEST_ID" NotOnOrAfter="2012-05-06T11:43:36Z"/>
        </saml:SubjectConfirmation>
    </saml:Subject>
    <saml:Conditions NotBefore="2012-05-06T11:38:36Z" NotOnOrAfter="2012-05-06T11:43:36Z">
        <saml:AudienceRestriction>
            <saml:Audience>http://serviceprovider.com</saml:Audience>
        </saml:AudienceRestriction>
    </saml:Conditions>
    <saml:AuthnStatement AuthnInstant="2012-05-06T11:38:36Z" SessionIndex="#SESSION_ID">
        <saml:AuthnContext>
            <saml:AuthnContextClassRef>
                urn:oasis:names:tc:SAML:2.0:ac:classes:PasswordProtectedTransport
            </saml:AuthnContextClassRef>
            <saml:AuthenticatingAuthority>http://idp.host.org</saml:AuthenticatingAuthority>
        </saml:AuthnContext>
    </saml:AuthnStatement>
    <saml:AttributeStatement>
        <!-- SAML Attribute with encrypted AttributeValue content -->
        <saml:Attribute Name="givenName" xmlns:ext="urn:oasis:names:tc:SAML:attribute:ext"
                        ext:OriginalIssuer="http://idp.host.org">
            ...
        </saml:Attribute>
    </saml:AttributeStatement>
</saml:Assertion>
```

Fig. 10 SAML Assertion including an encrypted AttributeValue

fore, we are using SAML as the underlying identity protocol, and in particular, we will describe how our model fits within the Authentication Request protocol, explained in Section 3.3.1. In our case, this protocol permits the service provider to request an assertion about the identity of the employee, including encrypted attributes.

Figure 9 shows the protocol interaction that takes place between the four entities involved: the employee as a user, the service provider as requester, the cloud provider as a proxying identity provider, and the host organization as authentication provider. The full sequence, assuming there is no security context for the employee at the service provider, works as follows:

1. The employee tries to access a protected resource offered by the service provider.
2. A discovery process occurs between the service provider and the user to find out the location of the identity provider; this process is out of the scope of both SAML and our system. The simplest option is that the user simply provides the location of the identity provider; a more complex option could be to integrate SAML with other discovery mechanisms, such as Yadis or XRI.
3. The service provider creates a SAML `AuthnRequest`.
4. The user agent gets redirected to the cloud identity provider through an HTML form, as explained in the SAML HTTP POST binding [40].
5. The cloud identity provider receives the authentication request. As we have mentioned before, BlindIdM uses a hybrid IDaaS approach, where the authentication remains at the premises of the host organization; as a consequence, the cloud provider must devolve the authentication to the host organization using the proxying mechanisms defined in the SAML Authentication Request protocol. In this case, the cloud identity provider issues a second authentication request, addressed to the host organization.
6. The user agent gets redirected to the host organization.
7. The employee is authenticated to the host organization. The authentication method is beyond of the scope of this work; for simplicity, we will assume that a password-based method is used.
8. The cloud provider constructs a SAML `Response` that responds to the second authentication request, and that conveys the authentication result and the identifier of the employee.
9. Once again, the user agent gets redirected to the cloud identity provider, delivering the authentication response from the host organization.
10. The cloud identity provider gets the encrypted attributes associated to the provided employee's identifier and, using the proper re-encryption key $r_{H \to SP}$ (obtained during the previous phase), proceeds to re-encrypt the ciphered attributes; actually, for each attribute a, it only has to re-encrypt the ciphered symmetric key K_a. Let $c_a = (c_{a,1}, c_{a,2}, c_{a,3})$ be one of the outsourced attributes, and R be the AFGH re-encryption function; then, the re-encrypted attribute c'_a is:

    $$c'_a = (c_{a,1}, R(r_{H \to SP}, c_{a,2}), c_{a,3}))$$

    Once the attributes are re-encrypted, the cloud provider issues a SAML `Assertion` that includes the encrypted attributes within an attribute statement, as well as the authentication statement from the host organization (obtained in the previous step); an `AuthenticatingAuthority` element is also included in the authentication statement, which references the authentication provider (in this case, the host organization). The cloud provider then encloses the assertion in the authentication response. Figure 10 shows the SAML `Assertion`.
11. Once again, the user agent gets redirected to the service provider, delivering the authentication response from the cloud identity provider, which includes the re-encrypted attributes.
12. The service provider verifies the authentication response and extracts the encrypted attributes from the assertion. Now, it simply has to decrypt the ciphertexts using its private key s_{SP}. Lets assume that $c'_a = (c'_{a,1}, c'_{a,2}, c'_{a,3})$ is one of the received attributes, D_S is the symmetric decryption algorithm, and D_1 is the AFGH first-level decryption function; then, the decrypted attribute a' is:

    $$\begin{aligned} a' &= (c'_{a,1}, D_S(D_1(s_{SP}, c'_{a,2}), c'_{a,3})) \\ &= (a.metadata, D_S(K_a, c'_{a,3})) \\ &= (a.metadata, a.value) \end{aligned}$$

3.5 Analysis

The main requirement of our model is to achieve end-to-end confidentiality for the identity information, enabling it to be stored in the cloud and managed blindly. Taking into account the trust model that we are using, that is, honest-but-curious providers, we argue that this requirement is fulfilled since the identity provider does not have access at any moment to the decryption keys.

The cloud provider has control only over the re-encryption process, but requires re-encryption keys that are generated by the host organization using its private key. As we have stated before, the re-encryption key, apart from making the re-encryption of ciphertexts possible, also acts as an authorization token, since it is generated by the host organization to give access to service providers to the identity information of its employees. Since we are assuming a honest-but-curious cloud provider, we can assume that it will remove the re-encryption key when asked. Ideally, temporary re-encryption keys that are valid only for a specific period of time would be used, so keys should not be valid if used at any other moment; this way, the re-encryption process could be cryptographically controlled by the host organization. To date, we have not seen any proxy re-encryption scheme that deals with re-encryption keys that are valid for a particular period of time only. In [32], the authors propose a temporary proxy re-encryption scheme that restricts both the re-encryption key generation and encryption processes to the same time period; however, this is not what we are looking for, since in our model the re-encryption key generation process will probably happen after the encryption, in a later time period.

It is important to note that our proposal does not require any change in the SAML framework, as we are respecting its protocols and constructions. Our model requires just a few extension points in the cloud identity provider and service provider, in order to re-encrypt and decrypt the attributes, respectively. We provide explicit SAML constructions that reflect how to realize our system using this framework.

From a practical point of view, it is also crucial to determine whether our proposal is economically feasible. Most of cryptography-based proposals only provide theoretical analysis of security and complexity, but do not tackle the economic viability. A detailed economic assessment about the use of proxy re-encryption in a cloud setting is presented in [41], where the authors estimate the cost of proxy re-encryption operations in USD cents; these expenses are a consequence of the incurred cost of the cryptographic computations in a cloud environment. For a detailed description of this analysis and the rationale behind these estimations, we refer the reader to this text. The costs of these operations, presented in Table 1, are the same as in our system since we are using the same proxy re-encryption scheme. For instance, it can be seen that the re-encryption operation, which is the one executed by the cloud provider, has an estimated cost of 4.79E-04 USD cents; in other words, the cloud identity provider can perform approximately 2087 re-encryptions for one USD cent. From these figures we can conclude that the cryptographic overhead is reasonable, as it permits an IDaaS system to serve thousands of encrypted attributes for a few cents, considering the costs that an organization could incur in the case of a disclosure or security breach; althought these costs are difficult to estimate due to their business and legal nature, at the very least such incidents would have a negative impact with regard to reputation and loss of customers. As an illustrative example, let us suppose that the IDaaS system receives a million attribute requests per day, which implies a million re-encryptions per day. This represents an additional expense of approximately 2000 USD on a yearly basis, since the cost for a re-encryption is 4.79 E-04 cents. We think these figures are reasonable for an average-sized organization, but ultimately it would depend on the savings from outsourcing their identity services to the cloud.

Table 1 Costs for the main cryptographic operations

Operation	Time (ms)	Cost (cents)	#ops/cent
Encryption	23.31	4.34E-04	2304
Re-encryption	90.09	4.79E-04	2087
Decryption	14.28	5.70E-04	1755

BlindIdM also complies with our vision of Identity Management as a Service, where competences are distributed in the cloud ecosystem. This affirmation is reflected in the hybrid approach we are using, where authentication is held on-premise by the host organization and data protection mechanisms (in our case, based in proxy re-encryption) are distributed in all the entities involved. Furthermore, the identity management protocol is followed by all the entities, and the host organization acts as the authoritative source of information. Figure 11 depicts this distribution of competences. Additionally, our model could easily support other variations, such as the externalization of the storage of encrypted attributes to a specialized cloud storage provider; the original cloud identity provider could then act as a mere intermediary in the identity interactions and would not have to own dedicated storage facilities.

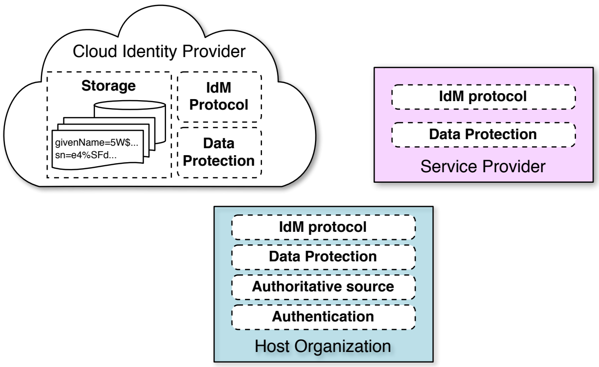

Fig. 11 Distributed competences in BlindIdM

3.6 Discussion: Privacy and Confidentiality

Privacy is a vague concept that on many occasions is used as an umbrella term, including other related concepts, such as confidentiality, unlinkability, anonymity, etc. For example, the term *privacy* is often used in the context of the unlinkability property; however, as we have already mentioned, unlinkability is not what we are addressing in this paper, but rather data confidentiality, as one of our goals is that identity information remains inaccessible to attackers, unauthorized entities, and even the cloud provider itself.

According to [42], *privacy* is defined as *"the right of an entity (normally a person), acting in its own behalf, to determine the degree to which it will interact with its environment, including the degree to which the entity is willing to share information about itself with others"*, while *data confidentiality* is defined as *"the property that data is not disclosed to system entities unless they have been authorized to know the data"*.

Taking these definitions into consideration, we argue that our system is *privacy-preserving* because it provides a data confidentiality service, in our case through the use of proxy re-encryption. This service cryptographically protects users' identity information and controls and limits the disclosure of private information with regard to the cloud provider, who acts as an intermediary in the identity management interactions.

4 Related Work

The problem of privacy in identity management is a widely studied subject. However, the data confidentiality aspects of privacy are seldom tackled. In [41], the authors propose a user-centric IDaaS system based in OpenID and proxy re-encryption. Although conceived as a proof of concept, this is the first work that achieves blind processing of identity information; however, trust issues arise as OpenID does not provide proper mechanisms for establishing trust. This proposal is useful for user-centric scenarios where service providers can fully trust end-users without the identity provider being able to assert any claim. One interesting aspect of this work is an economic assessment of the viability of the proposal; in rough numbers, they estimate that the cost for 2000 operations (i.e., encryptions, re-encryptions or decryptions) is 1 USD cent. This assessment is very relevant to our proposal, since we are using the same proxy re-encryption scheme, and therefore, the economic assessment also holds true for our case.

In [43], the authors propose a solution based on the deployment of active bundles in the cloud provider. An active bundle is a mobile agent, in this case a virtual machine, which contains the identity information of the user and that is protected by cryptographic means. Every time an operation involves the use of identity information, the cloud provider interacts with an active bundle to retrieve this information. However, this approach seems to be impractical because of the large overhead that the use of a large container for data (a VM) introduces. Moreover, the proposal does not detail any procedure to transport these active bundles to the cloud in an efficient manner.

Another proposal, based on the use of sticky policies and trusted computing, is presented in [7]. This paper presents an interesting approach where information, together with a specific policy that should be enforced in order to disclose the data, is obfuscated before leaving the users' domain. In this approach, a trusted authority is in charge of giving the receiver the means to deobfuscate the information, after verifying that the receiver complies with its associated policy; trusted computing is used to ensure the integrity of both software and hardware environments of the receiver. However, this work focuses on the direct sharing of information, which makes it unusable in an identity management setting, where an identity provider is used as an intermediary and must somehow manage this information.

Much work has been carried out regarding unlinkability of users with respect to the other entities involved in the identity management processes. For example, in [44] the authors describe the application of anonymous credentials to enhance privacy in identity management systems; in this case, the approach they propose is aimed to credential-focused systems, which is not our case. In [45], the authors present PseudoID, a model for private federated login that achieves unlinkability of users to visited sites. To this end, a blind signature service participates during the generation of an access token that is handed to the identity provider; this access token consists of a pseudonym and a secret value, that are both used to anonymously authenticate the user. Although this work presents an interesting contribution to privacy-enhanced identity providers, it is centered on the unlinkability aspects of the authentication of users. Moreover, this model is not suitable for maintaining users' information in the identity providers, since the providers are unable to correlate users to their pseudonyms.

With regard to the intersection of identity management, privacy and cloud computing, there has also been some research done. In [46], the authors propose SPICE, an identity management system for cloud environments whose main goal is to preserve users' privacy. SPICE satisfies a set of properties that the authors claim an identity management system in the cloud should fulfill, such as unlinkability and delegatable authentication. In order to accomplish this, SPICE uses a re-randomizable group signature scheme. However, the goal of SPICE is not the same as ours, since we are not tackling unlinkability, but data confidentiality. In [47], a privacy-preserving identity management system for cloud environments is presented; this system is based on zero-knowledge proofs that allow the user to prove the knowledge of a set of attributes without revealing their value. The problem of heterogeneity of attributes representation is also addressed in this work by using ontology mapping techniques. However, the authors do not tackle the privacy issues that are the main concern of our work, since in their setting, identity providers store in clear the values of the attributes of the users.

The use of cryptography for securing the cloud is a current research hot topic [48]. For example, in [49], the authors describe a high-level architecture that enables to build a secure cloud storage service from an untrusted cloud provider combining three recent cryptographic techniques, namely searchable encryption, proofs of storage and attribute-based encryption. Their proposal stands at a high level and does not include details regarding specific cryptographic primitives or procedures, but offers an interesting initial approach to the cloud storage problem through the use of cryptography.

5 Conclusions and Future Work

In this paper, we propose a solution to the problem of privacy for Identity Management as a Service. IDaaS is a recent trend, powered by cloud computing technologies, that allows companies and organizations to benefit from outsourcing identity management processes. The reduction of costs and time-consuming tasks associated with managing identity services are the main reasons behind this externalization. However, as is the case for other cloud-based services, there is much concern regarding the inversion of the control of the data, as users lose almost all control over their data.

We propose BlindIdM, an IDaaS system that guarantees user's privacy and control even when data storage and processing is performed by untrusted clouds. In particular, the main contribution of this paper is the construction of a privacy-preserving IDaaS system, where the cloud identity provider is able to offer an identity information service without knowing the actual personal information of the users. Our system uses SAML 2.0 as the underlying identity management protocol and proxy re-encryption as a means for achieving blind handling of identity information; this way, the cloud provider transforms encrypted attributes by the host organization into ciphertexts for the service provider, without being able to read their content during this process. In addition, we use standard SAML constructions for conveying this information. We believe that the approach presented in this paper opens up new possibilities regarding privacy in the field of identity management.

In this paper we have also discussed a new perspective of identity management in the cloud that unleashes its full potential by making use of the natural efficiency of distributed cloud-based services. We identify a set of competences involved in the management of identity information, in such a way that they can be distributed and orchestrated throughout the ecosystem of the cloud in order to achieve an appropriate model for IDaaS.

With regard to future work, we plan to deploy a prototype of our system in a real cloud setting, such as Amazon EC2 or Google AppEngine; in addition, more recent proxy re-encryption schemes could be used in order to provide more efficiency and security. We are also investigating other identity management technologies for the cloud that could be of use to extend our system, such as SCIM [50].

Acknowledgements This work was partly supported by the projects FISICCO (P11-TIC-07223) and ARES (CSD2007-00004). The first author has been funded by a FPI fellowship from the Junta de Andalucia through the project PISCIS (P10-TIC-06334).

References

1. John Hermans and Mike Chung. KPMG's 2010 Cloud Computing Survey. Technical report, KPMG, 2010.
2. Security guidance for critical areas of focus in cloud computing, version 3.0. Technical report, Cloud Security Alliance, 2011.
3. Cisco global cloud networking survey. Technical report, Cisco, 2012.
4. Jay Heiser and Mark Nicolett. Assessing the security risks of cloud computing. Technical report, Gartner Inc., 2008.
5. Top threats to cloud computing, version 1.0. Technical report, Cloud Security Alliance, 2010.
6. The Notorious Nine: Cloud Computing Top Threats in 2013. Technical report, Cloud Security Alliance, 2013.
7. M. Casassa Mont, S. Pearson, and P. Bramhall. Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services. In *Proc. 14th International Workshop on Database and Expert Systems Applications*, pages 377–382. IEEE, 2003.
8. R. Dhamija and L. Dusseault. The seven flaws of identity management: Usability and security challenges. *Security & Privacy, IEEE*, 6(2):24–29, 2008.
9. M. Hussain. *The Design and Applications of a Privacy-Preserving Identity and Trust-Management System.* PhD thesis, School of Computing, Queen's University, 2010.
10. OASIS Security Services TC. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.

11. Shibboleth Consortium. Shibboleth. `http://shibboleth.net/`.
12. OASIS Web Services Federation TC. Web Services Federation Language (WS-Federation) Version 1.2, 2009.
13. OASIS Security Services TC. Glossary for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.
14. E. Maler and D. Reed. The venn of identity: Options and issues in federated identity management. *Security & Privacy, IEEE*, 6(2):16–23, 2008.
15. Microsoft. Windows Azure Active Directory. `http://www.windowsazure.com/en-us/home/features/identity/`.
16. CA Technologies. CA CloudMinder Identity Management. `http://www.ca.com/us/cloudminder-identity-management`.
17. S. Pearson and A. Benameur. Privacy, security and trust issues arising from cloud computing. In *2nd IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 693–702. IEEE, 2010.
18. S. Clauß and M. Köhntopp. Identity management and its support of multilateral security. *Computer Networks*, 37(2):205–219, 2001.
19. Sabrina De Capitani di Vimercati, Sara Foresti, and Pierangela Samarati. Managing and accessing data in the cloud: Privacy risks and approaches. In *Risk and Security of Internet and Systems (CRiSIS), 2012 7th International Conference on*, pages 1–9. IEEE, 2012.
20. E.U. Comission. Council Directive 95/46/EC: On the protection of individuals with regard to the processing of personal data and on the free movement of such data, 1995.
21. Scott Shane and John F. Burns. U.S. Subpoenas Twitter Over WikiLeaks Supporters. The New York Times, January 8, 2011.
22. U.S. Congress. Uniting and strengthening america by providing appropriate tools required to intercept and obstruct terrorism act, 2001.
23. U.S. Congress. Health insurance portability and accountability act, 1996.
24. W. Kuan Hon, Christopher Millard, and Ian Walden. The problem of 'personal data' in cloud computing: what information is regulated? - the cloud of unknowing. *International Data Privacy Law*, 1(4):211–228, 2011.
25. Geoffrey A. Fowler, Devlin Barrett, and Sam Schechner. U.S. shuts offshore file-share 'locker'. The Wall Street Journal, January 20, 2012.
26. Certivox. PrivateSky. `http://privatesky.me/`.
27. CipherCloud. CipherCloud Gateway. `http://www.ciphercloud.com/`.
28. Y. Chen and R. Sion. On securing untrusted clouds with cryptography. In *Proceedings of the 9th annual ACM workshop on Privacy in the electronic society*, pages 109–114. ACM, 2010.
29. Stefanos Gritzalis. Enhancing web privacy and anonymity in the digital era. *Information Management & Computer Security*, 12(3):255–287, 2004.
30. J. Camenisch and E. Van Herreweghen. Design and implementation of the idemix anonymous credential system. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 21–30. ACM, 2002.
31. OASIS Security Services TC. Metadata for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.
32. G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. In *Proceedings of the 12th Annual Network and Distributed System Security Symposium*, pages 29–44, 2005.
33. M. Blaze, G. Bleumer, and M. Strauss. Divertible protocols and atomic proxy cryptography. *Advances in Cryptology—EUROCRYPT'98*, pages 127–144, 1998.
34. M. Green and G. Ateniese. Identity-based proxy re-encryption. In *Applied Cryptography and Network Security*, pages 288–306. Springer, 2007.
35. C.K. Chu and W.G. Tzeng. Identity-based proxy re-encryption without random oracles. *Information Security*, pages 189–202, 2007.
36. R. Canetti and S. Hohenberger. Chosen-ciphertext secure proxy re-encryption. In *Proceedings of the 14th ACM conference on Computer and communications security*, pages 185–194. ACM, 2007.
37. B. Libert and D. Vergnaud. Unidirectional chosen-ciphertext secure proxy re-encryption. *Information Theory, IEEE Transactions on*, 57(3):1786–1802, 2011.
38. G. Ateniese, K. Benson, and S. Hohenberger. Key-private proxy re-encryption. *Topics in Cryptology–CT-RSA 2009*, pages 279–294, 2009.
39. W3C. XML Encryption Syntax and Processing Version 1.0. W3C Recommendation, W3C, 2002. http://www.w3.org/TR/xmlenc-core/.
40. OASIS Security Services TC. Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0, 2005.
41. David Nuñez, Isaac Agudo, and Javier Lopez. Integrating OpenID with proxy re-encryption to enhance privacy in cloud-based identity services. In *Cloud Computing Technology and Science (CloudCom), 2012 IEEE 4th International Conference on*, pages 241–248. IEEE, 2012.
42. R. Shirey. Internet Security Glossary, Version 2. RFC 4949 (Informational), August 2007.
43. P. Angin, B. Bhargava, R. Ranchal, N. Singh, L.B. Othmane, L. Lilien, and M. Linderman. An entity-centric approach for privacy and identity management in cloud computing. In *29th IEEE Symposium on Reliable Distributed Systems*, pages 177–183, 2010.
44. Claudio A Ardagna, Jan Camenisch, Markulf Kohlweiss, Ronald Leenes, Gregory Neven, Bart Priem, Pierangela Samarati, Dieter Sommer, and Mario Verdicchio. Exploiting cryptography for privacy-enhanced access control: A result of the PRIME project. *Journal of Computer Security*, 18(1):123–160, 2010.
45. Arkajit Dey and Stephen Weis. PseudoID: Enhancing privacy in federated login. In *Hot Topics in Privacy Enhancing Technologies*, pages 95–107, 2010.
46. S. Chow, Y.J. He, L. Hui, and S. Yiu. SPICE–simple privacy-preserving identity-management for cloud environment. In *Applied Cryptography and Network Security*, pages 526–543. Springer, 2012.
47. E. Bertino, F. Paci, R. Ferrini, and N. Shang. Privacy-preserving digital identity management for cloud computing. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 32(1):21–27, 2009.
48. Isaac Agudo, David Nuñez, Gabriele Giammatteo, Panagiotis Rizomiliotis, and Costas Lambrinoudakis. Cryptography goes to the cloud. In *Secure and Trust Computing, Data Management, and Applications*, pages 190–197. Springer, 2011.
49. S. Kamara and K. Lauter. Cryptographic cloud storage. *Financial Cryptography and Data Security*, pages 136–149, 2010.
50. System for cross-domain identity management. `http://www.simplecloud.info/`.