User-Centric Identity as a Service-Architecture for eIDs with Selective Attribute Disclosure

Daniel Slamanig
Graz University of Technology (IAIK)
daniel.slamanig@tugraz.at

Klaus Stranacher
E-Government Innovation Center (EGIZ)
klaus.stranacher@egiz.gv.at

Bernd Zwattendorfer
E-Government Innovation Center (EGIZ)
bernd.zwattendorfer@egiz.gv.at

ABSTRACT

Unique identification and secure authentication of users are essential processes in numerous security-critical areas such as e-Government, e-Banking, or e-Business. Therefore, many countries (particularly in Europe) have implemented national eID solutions within the past years. Such implementations are typically based on smart cards holding some certified collection of citizen attributes and hence follow a client-side and user-centric approach. However, most of the implementations only support all-or-nothing disclosure of citizen attributes and thus do not allow privacy-friendly selective disclosure of attributes. Consequently, the complete identity of the citizen (all attributes) are always revealed to identity providers and/or service providers, respectively. In this paper, we propose a novel user-centric identification and authentication model for eIDs, which supports selective attribute disclosure but only requires minimal changes in the existing eID architecture. In addition, our approach allows service providers to keep their infrastructure nearly untouched. Latter is often an inhibitor for the use of privacy-preserving cryptography like anonymous credentials in such architectures. Furthermore, our model can easily be deployed in the public cloud as we do not require full trust in identity providers. This fully features the Identity as a Service-paradigm while at the same time preserves citizens' privacy. We demonstrate the applicability of our model by adopting to the Austrian eID system to our approach.

Categories and Subject Descriptors

K.6.5 [**Management of Computing and Information Systems**]: Security and Protection—*Authentication*; K.4.1 [**Computers and Society**]: Public Policy Issues—*Privacy*

Keywords

Identity management; authentication; privacy; selective attribute disclosure; cloud computing; public cloud; Austrian eID; citizen card

SACMAT '04 London, Ontario Canada
Copyright 2014 ACM 978-1-4503-2939-2/14/06 ...$15.00.
http://dx.doi.org/10.1145/2613087.2613093.

1. INTRODUCTION

Identification and authentication play a vital role in numerous fields of application. Particularly, in areas such as e-Government, e-Banking, or e-Business these processes are highly security critical and complex. In order to cope with this complexity, various identity management systems such as SAML[1] or OpenID[2] have already evolved [2]. The main functions of identity management systems, besides secure and easy management of identities and corresponding attributes, are the handling of identification and authentication of users. Identity management systems usually target different horizons, e.g., the management of employees of a company or organization, or the management of the population of a whole country. For supporting the latter, many countries have already rolled-out national eID solutions. The aim of such solutions is to use authentic and qualified citizen attributes in sensitive electronic processes, where a high level of security is required.

1.1 Identity Management Models

An identity management system involves different stakeholders. Typically, *users* first need to identify themselves and authenticate at an *identity provider* before being able to access protected services and resources of a *service provider*. Thereby, the identity provider transfers authentic identity data of the user to the service provider in a structured form (either directly or via the user). Over time, different identity models have emerged, having differences in terms of user control or identity data storage location [20, 12, 13].

In the *central identity model*, user and identity data are stored centrally at the identity provider. Identity data provisioning is usually carried out by user registration at the identity provider prior first authentication. During an authentication process, identity data are transferred from the identity provider to the service provider. However, in this model the user usually is not aware which kind of data are actually stored in the identity provider's repositories and to whom it is released.

In the *federated identity model*, user and identity data are stored distributed across different identity providers. These data can be linked and exchanged if appropriate trust relationships exist amongst the identity providers. Trust relationships are usually established on organizational level whereas they are enforced on technical level. As above, in this model the user is usually not aware of the data stored

[1] Security Assertion Markup Language, `http://saml.xml.org`

[2] `http://openid.net`

at the identity provider and to whom the identity provider releases the data.

Finally, in the *user-centric identity model* identity data are directly stored in the user's domain, usually on a secure token such as a smart card. Identity data are only transferred to an identity provider or a service provider if the user explicitly gives her consent to do so. Hence, the user always remains in the sole control of her data. Many countries rely on this model for their national eID solutions. For instance, Austria stores citizens' identification data (e.g., unique identifier, name, data of birth) within a special data structure on the Austrian citizen card, the official eID in Austria [14]. This special data structure is signed by a trusted public authority to ensure integrity and authenticity of the data. Other countries such as Belgium, Estonia, Italy, or Spain include identity data (e.g., unique identifier and name) for their eID directly in an X.509 certificate, which is used for identification and authentication. Many other European countries follow the same or similar approaches for modeling electronic identities of their citizens. Details on the individual approaches can be found in [17, 7, 23].

1.2 Contribution

While all these national eID solutions are able to provide authentic and qualified identity attributes to service providers in a user-centric manner, they nearly all lack in selective disclosure possibilities as only the full eID can be revealed during an authentication process [18]. To close this gap and to better protect citizens' privacy as demanded by the EU data protection directive [8], a couple of approaches exist. However, in all these existing approaches several disadvantages can be identified limiting their practicality.

In this paper we propose a novel and practical identification and authentication model to be applied for eIDs, which keeps the advantages of user-centricity but allows for selective disclosure possibilities to better protect citizens' privacy compared to existing national eID solutions. In particular, it supports selective disclosure such that any service provider can still verify that the subset of revealed data has been certified by a trusted authority. Furthermore, our model can be easily deployed in semi-trusted environments such as the public cloud to fully benefit from their scalability and elasticity advantages. Semi-trusted thereby means *honest but curious*, i.e., such a party performs all tasks correctly, but processed or stored data may be leaked either due to adversarial insiders or security breaches. This is a reasonable adversarial model for public cloud providers. Finally, to demonstrate the applicability of our model, we apply it to the Austrian eID system.

1.3 Paper Outline

The paper is structured as follows. In Section 2 we overview approaches and existing systems enabling selective disclosure capabilities for electronic identities. In Section 3 we describe our abstract model, which enables user-centricity and selective disclosure for eIDs in a practical manner at the same time. After that, we apply our model to the current Austrian eID system in Section 4. Finally, we discuss our model – compared to existing approaches – based on selected criteria in Section 5.

2. SELECTIVE DISCLOSURE

To better protect citizens' privacy and to enable selective disclosure of authentic and qualified attributes of eIDs, different approaches have been proposed. In this section, we briefly review them. We distinguish between approaches based on the trust assumptions for the identity provider (identity provider is trusted or semi-trusted).

2.1 Trusted Identity Provider

A typical example for such an approach is the new German eID card [16]. The German eID card follows a user-centric approach enabling minimum data disclosure to the service provider. Each service provider gets issued a so-called access certificate which enables controlled access to a subset of identity data stored on the eID card. To facilitate the integration of eID functionality for service providers, an eID service acts as intermediary between the service provider and the eID card. The eID service manages the communication with the eID card and provides identity data from the eID to the service provider. While the citizen is able to select the data to be disclosed to the service provider on client-side through a client middleware, data are still available in plain form at the eID service and identity data not relevant for the eID service may be revealed to it. Hence, the eID service must be fully trusted in terms of privacy and data protection.

Another example for this approach is STORK[3] [15], which deals with eID federation and eID interoperability across EU countries. One interoperability model of STORK foresees a central gateway per country, which on the one side connects to the national eID infrastructure and, on the other side, transfers authentic identity data to national gateways of other countries in cross-border authentication scenarios. This national gateway only transfers the amount of data which has been requested by a foreign service provider and the citizen has given consent to. Thereby, the citizen gives her consent on the national gateway, hence on server-side. This implies that – depending on the national eID infrastructure – the national gateway may still get the full identity of the citizen before it releases only a subset of the identity data to a foreign country gateway. As above, the user can also not be sure that the correct subset of identity data is revealed. Therefore, the central gateway again needs to be fully trusted in terms of data protection and privacy.

2.2 Semi-Trusted Identity Provider

Privacy and data protection are particular issues in the (public) cloud, where cloud providers are able to inspect stored data if it is not encrypted. Thus, cloud providers can be seen as semi-trusted. This issue also holds for Identity as a Service (IDaaS) solutions, where the identity provider typically is operated in a public cloud.

To bypass this issue in IDaaS scenarios, Nuñez et al. [19] propose an approach to extend the OpenID protocol by using proxy re-encryption (cf. Section 3.3). Thereby, identity data are stored encrypted by the user at an OpenID provider, which is operated in a public cloud. If a service provider requires identity data for authentication, the identity data are re-encrypted for the service provider by the OpendID provider using a re-encryption key provided by

[3] Secure Identities Across Borders Linked, http://www.eid-stork.eu

the user. Consequently, the OpenID provider does not learn any of the user's identity data (attributes) in plaintext.

Anonymous credential systems are a typical example for a setting where the identity provider is not fully trusted. Thereby, the credential is issued in a way that issuing and showing of a credential cannot be related. Only claims of user attributes are transmitted to the service provider without revealing the full user identity. The generation of claims is usually carried out on client-side. Well known anonymous credential systems are for instance the multi-show (unlinkable showings) Idemix [6] and one-show (linkable showings) U-Prove [4] systems. While anonymous credential systems are valuable means to support selective disclosure of attributes in authentication scenarios, they still lack in practicability as the underlying cryptographic technologies are computationally expensive and put a lot of load onto the client-side software and smart card [3]. Furthermore, easy integration of anonymous credentials into existing identity protocols such as SAML or OpenID is still lacking. However, projects such as PRIME[4], PrimeLife[5], or ABC4Trust [22] put lots of efforts in cryptographic abstraction and standardization in order to make future integration easier. Nevertheless, if adopting such systems at the moment, service providers may have to significantly change their existing identification and authentication infrastructure.

Another approach for semi-trusted identity providers has been proposed by Zwattendorfer and Slamanig [25]. In fact, they actually propose three different approaches, all relying on different cryptographic technologies. The first approach uses fully homomorphic encryption, which can be considered impractical at the present time. The second approach bases on anonymous credentials, which still lack in deployment. The third approach uses a combination of proxy re-encryption and redactable signatures, which they concluded to be the most practical one. The third approach has also been applied in [26].

3. THE PROPOSED MODEL

In this section, we propose a new user-centric identification and authentication model, which is particularly applicable for semi-trusted environments (in terms of data protection and privacy) such as the public cloud. Our model allows the usage of both server-side and client-side approaches for user data storage, while still putting users under full control of their data, i.e., providing selective disclosure in both approaches.

3.1 Motivation

Many qualified eIDs issued and rolled-out by various countries miss the feature of selective disclosure in current implementations. This means that users usually have to reveal their full identity during an identification and authentication process even if a service provider only needs a subset of the identity for providing a service. To bypass this issue, several approaches support selective disclosure via non-cryptographic means. An obvious solution, e.g., implemented by the German eID card or by STORK, is to simply trust the identity provider (cf. Section 2.1). As mentioned, the drawback of this solution is that the identity provider needs to be fully trusted, which makes a deployment in semi-trusted environments such as the public cloud impossible due to national data protection regulations.

In contrast to that, selective disclosure approaches such as anonymous credential systems allow the usage of semi-trusted environments. However, although a lot of research and work is going in that direction, they still lack in wide deployment and general acceptance. Other approaches supporting semi-trusted cloud environments, basically relying on proxy re-encryption, were proposed by Nuñez et al. [19] and Zwattendorfer and Slamanig [25]. While these approaches seem currently to be more practical compared to anonymous credential systems, they still have some drawbacks. The approach by Nuñez et al., for instance, is not applicable for qualified eIDs. In addition, a main drawback of the approach of Zwattendorfer and Slamanig is that their approach is strongly tailored to the Austrian eID system, which does not allow for general applicability. Furthermore, it requires quite cumbersome registration processes of identity providers and service providers. In addition, their approach is not fully user-centric as the user data are encrypted for a trusted third party and not the user herself.

The drawbacks of these existing solutions motivated us to design and develop an improved and more generic identification and authentication model for semi-trusted environments such as the public cloud.

3.2 Requirements

To be applicable for qualified eIDs, as currently in place in various countries, and to further extend their privacy capabilities, the model has to fulfill the following requirements. We distinguish between general requirements and requirements related to privacy.

General Requirements.

The following general requirements need to be fulfilled by our model:

Qualified and authentic identity data: The identity and the corresponding attribute data of a user has been verified and certified by a trusted authority and thus are of high quality. Every party that obtains (a subset of the) attribute data can verify that is has been certified by the trusted authority.

Semi-trusted identity providers: The model must be applicable to the semi-trusted identity providers approach, i.e., identity providers must not be considered as fully trustworthy.

Integration effort and complexity: The new model can be integrated into existing infrastructures without significant changes.

Privacy Requirements.

In addition, our model has to fulfill the following privacy requirements:

Privacy: The privacy of users' identity data must be preserved in the presence of an honest but curious (semi-trusted) identity provider, i.e., the identity provider must not learn anything about the identity attributes of a user.

[4] Privacy and Identity Management for Europe, `https://www.prime-project.eu`

[5] `http://primelife.ercim.eu`

User-centricity: The user always remains in full control over her identity data and solely the user can control which identity data will be revealed to other parties.

Selective Disclosure: The user can disclose only parts of her identity data to a service provider and the service provider does not learn anything about undisclosed attribute data.

3.3 Cryptographic Preliminaries

In this section we introduce the cryptographic building blocks that are required by our presented model. We provide an informal description here and refer the reader to Appendix A for a more formal description.

Proxy Re-Encryption.

A proxy re-encryption (PRE) scheme is a public key encryption scheme that allows a semi-trusted proxy to transform a ciphertext produced with a public key $\mathsf{pk}_{\mathsf{A}}^{\mathsf{PRE}}$ of party A into another ciphertext of another party B, which can be decrypted by the private key $\mathsf{sk}_{\mathsf{B}}^{\mathsf{PRE}}$, using a re-encryption key $\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}$. Thereby, the proxy neither gets access to the plaintext nor to the decryption keys. We rely on 1) non-interactive, 2) unidirectional, and 3) single-use schemes [1]. Basically, this means that 1) a re-encryption key $\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}$ can be computed by A using the private key $\mathsf{sk}_{\mathsf{A}}^{\mathsf{PRE}}$ and the public key $\mathsf{pk}_{\mathsf{B}}^{\mathsf{PRE}}$, 2) based on $\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}$ re-encryption from B to A is not possible, and 3) re-encryption can only be performed once to a given ciphertext (no transitivity).

Digital Signatures.

A digital signature scheme allows to produce a digital signature σ for a message M (using a private signing key $\mathsf{sk}^{\mathsf{DSS}}$) and given the signature σ, a message M and a public verification key $\mathsf{pk}^{\mathsf{DSS}}$ anyone can check whether the signature has been issued for M (integrity) by the holder of the corresponding signing key (authenticity).

Redactable Signatures.

A redactable signature (RS) scheme [11, 24] is a malleable digital signature scheme that supports the *removal* of parts of a signed message by *any* party, without requiring access to the signer's secret key and without invalidating the original signature. Consequently, parts of a signed message can be blacked out without invalidating the signature.

Blank Digital Signatures.

Blank digital signatures (BDS) [10] allow an *originator* to delegate the signing rights for a certain *template* T to a *proxy*, where T is a sequence of non-empty sets of bitstrings T_i. Such sets T_i are either called *fixed* or *exchangeable*, depending on the cardinality of the respective set, i.e., exchangeable elements contain more than one bitstring, whereas fixed elements contain exactly one bitstring. Based on such a delegation, the *proxy* is able to issue a signature on an instance of the template on behalf of the *originator*. Such a valid instance needs to include all fixed elements and one choice for each exchangeable element. With the instance signature at hand, anyone is able to verify the validity of the instance signature, i.e., if it is a valid instantiation of the template and the delegation, whilst the original template, i.e., the unused values of the exchangeable elements of the template, cannot be determined by a verifier. Blank digital signatures can also be used as redactable signature scheme. That means the originator issues a signature for a template, whereas the exchangeable parts consist of a certain value and a symbol representing the redacted (empty) string. Thus, the proxy is able to choose the redacted string to perform the redaction. Compared to core redactable signature schemes, blank digital signatures do not require an additional signing of the instance M (representing the redacted message) as the proxy signs it in course of producing the instance signature.

3.4 The Model

Figure 1 illustrates the new identification and authentication model for eIDs, which is applicable as an Identity as a Service model in semi-trusted environments. The following entities are involved in this model:

Registration authority: The registration authority (RA) is a trusted entity which issues qualified and authentic identity data to the user. The identity data can be either stored on client-side, e.g., a secure token, or on server-side at a trusted identity and/or attribute provider.

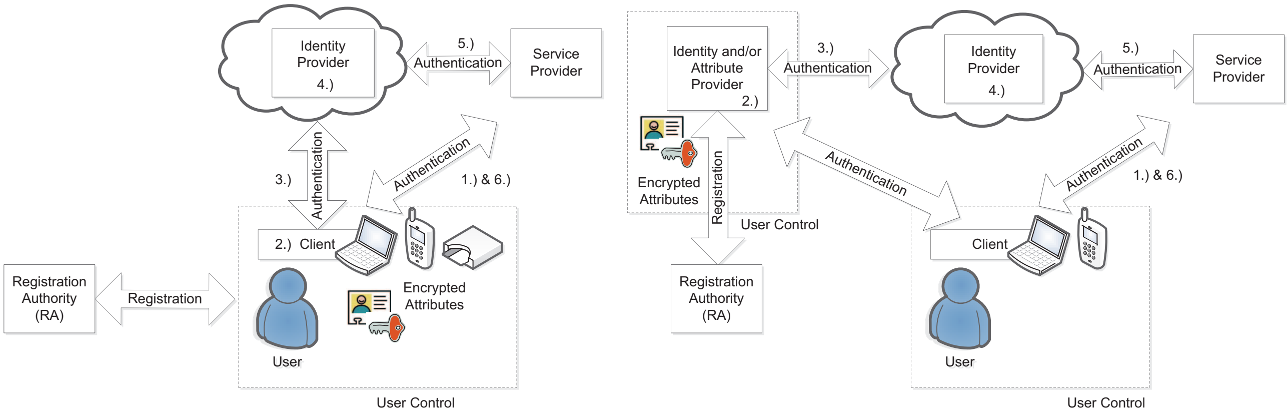
User: The user wants to access protected resources of a service provider. For gaining access, the user can reveal selected identity data issued from the registration authority.

Service provider: The service provider (SP) offers different resources or services which require qualified identification and authentication using eIDs.

Identity provider: The identity provider (IdP) is deployed in the cloud, meaning in a semi-trusted environment. The identity provider manages the identification and authentication process for the service provider and provides the service provider with asserted data via well-known protocols such as SAML or OpenID.

Identity and/or attribute provider: This entity holds qualified and authentic identity data of the user on server-side.

In this proposed model for eIDs, identity data will be encrypted (using a proxy re-encryption scheme) by the registration authority for the user in such a way that only the user is able to decrypt the data. Encrypting attributes only for the user gives the user sole control to her data. This form of encryption and selective disclosure fulfills the requirement of *user-centricity* on the one side, and the support of *semi-trusted identity providers* on the other side, as only encrypted data are provided to the identity provider. The user can give a subset of re-encrypted attributes to a service provider such that it can only be decrypted by this service provider (*selective disclosure*). Furthermore, the encrypted identity data are digitally signed by the trusted registration authority using a malleable signature scheme (redactable or blank digital signatures). Signing the data has basically two functions. First, the data are authentic and integrity can be assured as the data are signed by the trusted registration authority. This meets the requirement of *qualified and authentic identity data*. Second, by using a malleable signature scheme, we can guarantee that only required (encrypted) attributes can be disclosed to the service provider without



(a) Client-side model

(b) Server-side model

Figure 1: A user-centric and privacy-preserving Identity as a Service Model for eIDs.

invalidating the signature of the trusted registration authority. Finally, the requirement of *easy integration into existing infrastructures* can be met by this model as existing identity protocols already support the transfer of encrypted data and digital signatures out-of-the-box.

In the following, we give details on the registration process and the identification and authentication process when applying this model. The registration process has to be conducted only once, whereas the latter must be performed for each access to a protected resource.

Registration Process: Qualified and authentic identity data issuance is carried out by a trusted third party, the registration authority. Data provisioning is done during an appropriate registration process between the user and the registration authority. Details of the registration process are out of scope of this model and are dependent on the respective eID approach. Nevertheless, registered identity data are encrypted for the user using a proxy re-encryption scheme and signed by the registration authority using a redactable or blank digital signature scheme. The encrypted and signed data can be either stored on a secure token featuring a client-side approach or on a remote server modeling a server-side approach. However, irrespective of the underlying approach, identity data are always provided by a trusted authority in an authentic and qualified manner. This allows our approach to be used for national eID solutions.

Identification and Authentication Process: Figure 1 also illustrates the identification and authentication process when applying this model. For better illustration, we assume the identity provider to be running in a public cloud to fully feature the Identity as a Service paradigm. Basically, the identity provider in the public cloud has three main responsibilities: 1) user authentication and verification of authenticity of encrypted identity data, 2) re-encrypting the identity data for the service provider, and 3) structuring and transferring identity data to the service provider. To illustrate the individual responsibilities, we briefly describe an authentication process using our model.

1. The user wants to access a protected resource at the service provider who requires authentication. Authentication is carried out by the identity provider. Hence, the user is forwarded there.
2. The user redacts all encrypted attributes which she does not want to disclose to the service provider. Depending on the underlying approach, this can be done on client-side or server-side. However, in both cases the user remains under sole control of her data. At the same time, the user also generates a re-encryption key based on a public key of the service provider and the user's private key.
3. The redacted identity data and the re-encryption key are sent to the identity provider in the cloud.
4. The identity provider verifies the authenticity and integrity of the identity data, i.e., the signature, and re-encrypts it for the service provider. Additionally, the identity data are structured accordingly for being transferred to the service provider.
5. The identity provider transfers the data to the service provider using appropriate existing identity protocols such as SAML or OpenID. To ensure authenticity and integrity, the identity provider signs the transferred data.
6. The service provider verifies the received data and decrypts the provided and asserted attributes. Based on the attribute values, the service provider either grants or denies access.

To securely and reliably support these functionality, we make some assumptions:

Assumptions: We assume that whenever we speak of public parameters or public keys, they are available in an authentic fashion, e.g., via a PKI. Furthermore, the channels between all parties provide confidentiality as well as authenticity, e.g., via the use of TLS.

4. APPLICATION TO THE AUSTRIAN EID SYSTEM

In this section we demonstrate the applicability of our model by applying it to an existing eID system, i.e., the Austrian eID system. Therefore, we briefly introduce the Austrian eID system and then illustrate the realization of the Austrian eID system when applying our proposed model.

4.1 The Austrian eID System

Unique citizen identification and secure authentication in Austria is based on the Austrian citizen card[6] [14], the official eID in Austria. Unique identification is based on a unique number, the so-called sourcePIN, which is wrapped in a special XML data structure, the so-called Identity Link (IDL), and stored on the citizen card. In more detail, the Identity Link includes the citizen's sourcePIN, first name, last name, date of birth, and a qualified certificate for creating digital signatures according to the EU Signature Directive [9] (the corresponding private key is also stored on the citizen card). To ensure authenticity and integrity of the Identity Link, it is digitally signed by the trusted SourcePIN Register Authority (SRA). We subsequently denote the Identity Link as $\mathsf{IDL} = ((A_1, a_1), \ldots, (A_m, a_m))$ being a sequence of identity attribute name A_i and value a_i pairs.

To preserve citizens' privacy, it is prevented by law (according to the Austrian e-Government Act [21]) to directly use the sourcePIN for identification at online applications. Therefore, the Austrian eID system implements a sector-specific identification model using domain-specific pseudonyms. These so called sector-specific PINs (ssPINs) are uniquely derived from the sourcePIN and ensure citizen unlinkability across multiple sectors.

In the following we briefly describe the registration and authentication process in the Austrian eID System. Figure 2 illustrates the involved entities and their interactions.

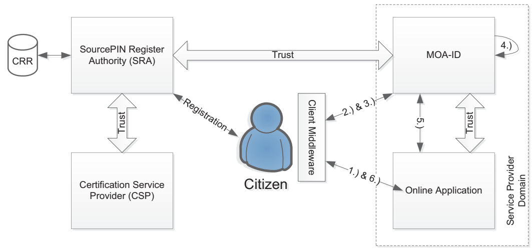


Figure 2: The Austrian eID system.

Registration Process: In order to activate an Austrian citizen card, citizens must prove their identity to the sourcePIN register authority (SRA). This can be done either in a registration office or via registered mail. The SRA finally creates the sourcePIN, the Identity Link, and the qualified certificate and issues the citizen card including these data to the citizen. More precisely, for this process the SRA relies on cryptographic key material provided by an certification service provider (CSP), which is accredited according to the EU Signature Directive.

Identification and Authentication Process: For facilitating the identification and authentication process using the Austrian citizen card at online applications, service providers usually rely on the open source module MOA-ID[7]. On the one side, this module manages the identification and authentication process with the citizen and, on the other side, provides citizen's identity data in a structured format to the online application. According to Figure 2, the identification and authentication process involves the following steps:

1. The citizen wants to access a protected resource, which requires citizen card authentication. The online application starts the authentication process and triggers MOA-ID.

2. First, MOA-ID reads the Identity Link from the citizen card through the client middleware and verifies it. This corresponds to the identification process.

3. Second, MOA-ID requests the citizen to create a qualified electronic signature[8] for authentication. The qualified electronic signature is verified by MOA-ID involving appropriate certificate revocation mechanisms (CRLs,OCSP) provided by the CSP.

4. MOA-ID derives the sourcePIN according to the domain the service provider is assigned to and thus creates a sector-specific PIN (ssPIN).

5. MOA-ID assembles a special data structure which includes the ssPIN and additional personal data of the citizen such as first name, last name, and date of birth. The assembled data structure, called *assertion*, follows the specification of SAML and is transmitted to the online application.

6. Based on the data received, the online application is able to provide the protected resource to the citizen.

4.2 Realization

The current deployment approach of the Austrian eID system foresees a local deployment of MOA-ID within each service provider's domain. Thereby, MOA-ID is fully trusted. While this approach has some clear benefits in terms of scalability and end-to-end security, a central approach may be still advantageous. However, a central approach probably lacks in terms of scalability. To overcome this issue, Zwattendorfer and Slamanig [25] proposed a move of MOA-ID into a public cloud, meaning into a semi-trusted environment. Nevertheless, the proposed approach has some clear disadvantages that our new model is able to overcome.

Subsequently, we present a use case of our user-centric and privacy-preserving Identity as a Service model for eIDs giving a concrete realization of the Austrian eID system. In our realization we make use of a proxy re-encryption scheme of Ateniese et al. [1], which omits the requirements of the service provider and MOA-ID to be registered at the SRA as in [25]. Additionally, we apply blank digital signatures [10] (used as a redactable signature scheme) for the Identity Link enabling the citizen to redact specific identity attributes out

[6] Currently, the Austrian citizen card is implemented as client-side approach using smart cards and as server-side approach involving the citizen's mobile phone.

[7] https://joinup.ec.europa.eu/software/moa-idspss

[8] Qualified electronic signature is a legal term for a digital signature which satisfies specific requirements according to the EU signature directive [9].

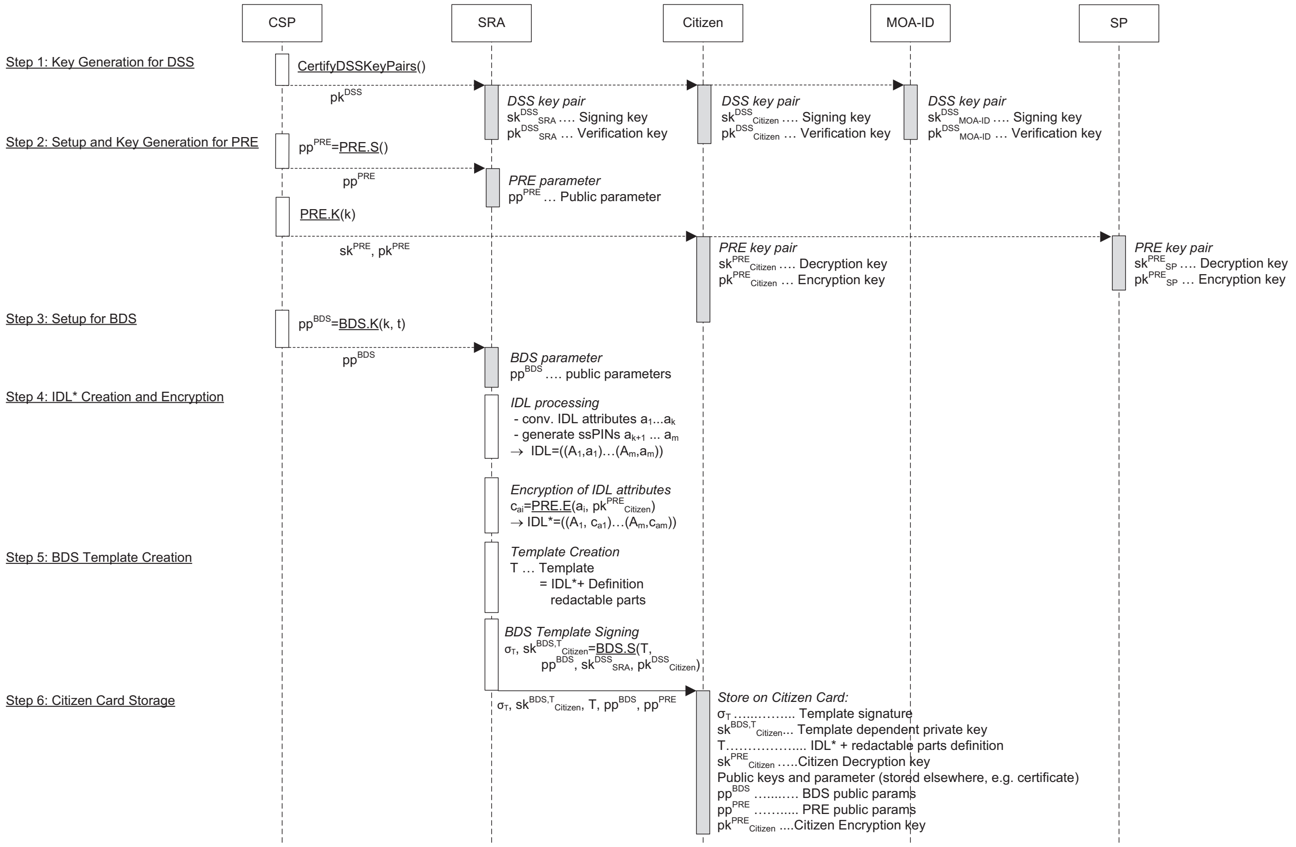

CSP
SRA
Citizen
MOA-ID
SP
Step 1: Key Generation for DSS
CertifyDSSKeyPairs()
pk^{DSS}
DSS key pair
sk^{DSS}_{SRA} …. Signing key
pk^{DSS}_{SRA} … Verification key
DSS key pair
$sk^{DSS}_{Citizen}$ …. Signing key
$pk^{DSS}_{Citizen}$ … Verification key
DSS key pair
$sk^{DSS}_{MOA\text{-}ID}$ …. Signing key
$pk^{DSS}_{MOA\text{-}ID}$ … Verification key
Step 2: Setup and Key Generation for PRE
pp^{PRE}=PRE.S()
pp^{PRE}
PRE parameter
pp^{PRE} … Public parameter
PRE.K(k)
sk^{PRE}, pk^{PRE}
PRE key pair
$sk^{PRE}_{Citizen}$ …. Decryption key
$pk^{PRE}_{Citizen}$ … Encryption key
PRE key pair
sk^{PRE}_{SP} …. Decryption key
pk^{PRE}_{SP} … Encryption key
Step 3: Setup for BDS
pp^{BDS}=BDS.K(k, t)
pp^{BDS}
BDS parameter
pp^{BDS} …. public parameters
Step 4: IDL* Creation and Encryption
IDL processing
- conv. IDL attributes $a_1...a_k$
- generate ssPINs a_{k+1} ... a_m
→ IDL=((A_1,a_1)…(A_m,a_m))
Encryption of IDL attributes
c_{ai}=PRE.E(a_i, $pk^{PRE}_{Citizen}$)
→ IDL*=((A_1, c_{a1})…(A_m,c_{am}))
Step 5: BDS Template Creation
Template Creation
T … Template
= IDL*+ Definition
redactable parts
BDS Template Signing
σ_T, $sk^{BDS,T}_{Citizen}$=BDS.S(T,
pp^{BDS}, sk^{DSS}_{SRA}, $pk^{DSS}_{Citizen}$)
Step 6: Citizen Card Storage
σ_T, $sk^{BDS,T}_{Citizen}$, T, pp^{BDS}, pp^{PRE}
Store on Citizen Card:
σ_T ………….. Template signature
$sk^{BDS,T}_{Citizen}$… Template dependent private key
T……………... IDL* + redactable parts definition
$sk^{PRE}_{Citizen}$ …..Citizen Decryption key
Public keys and parameter (stored elsewhere, e.g. certificate)
pp^{BDS} ………. BDS public params
pp^{PRE} ………. PRE public params
$pk^{PRE}_{Citizen}$ ....Citizen Encryption key

Figure 3: Sequence diagram of registration process.

of the Identity Link[9]. By using these technologies, our realization is easy integrable into the existing infrastructure and requires minimal changes on the service provider's side. In the following, we elaborate on the registration as well as on the identification and authentication process in detail.

Registration Process: From the user's and service provider's perspective the registration process does not change compared to the existing system. The required changes for applying our new model affect the creation of the data to be stored on the citizen card only and has to be done by the SRA as it is the case in the current approach. Figure 3 shows the sequence diagram of the registration process. In the following, we assume that the DSS secret keys are generated by the respective entities but for simplicity assume that the key pairs for the PRE scheme are generated by the SRA (in practice this can easily be done by every entity and only the public keys are certified by SRA or some CSP). The entire registration consists of the subsequent steps:

1. The CSP certifies the public signature verification key for the SRA ($\mathsf{pk}^{\mathsf{DSS}}_{\mathsf{SRA}}$), the Citizen ($\mathsf{pk}^{\mathsf{DSS}}_{\mathsf{Citizen}}$), and MOA-ID ($\mathsf{pk}^{\mathsf{DSS}}_{\mathsf{MOA-ID}}$).

2. The CSP generates the public parameter $\mathsf{pp}^{\mathsf{PRE}}$ for the PRE and publishes it. In addition, the CSP generates the PRE key pair for the Citizen ($\mathsf{sk}^{\mathsf{PRE}}_{\mathsf{Citizen}}, \mathsf{pk}^{\mathsf{PRE}}_{\mathsf{Citizen}}$), which will be stored on the Citizen's citizen card in step 6. Finally, the PRE key pair for the service provider ($\mathsf{sk}^{\mathsf{PRE}}_{\mathsf{SP}}, \mathsf{pk}^{\mathsf{PRE}}_{\mathsf{SP}}$) is generated and given to the service provider.

3. The CSP generates the public parameter $\mathsf{pp}^{\mathsf{BDS}}$ for the BDS scheme and publishes it.

4. The SRA creates a modified Identity Link IDL^* based upon the original IDL attributes. This IDL^* includes the attributes of IDL (e.g., name, date of birth, etc.) and *all* domain-specific pseudonyms (ssPINs) for all public sectors. Furthermore, the SRA encrypts these identity attributes for the Citizen using $\mathsf{pk}^{\mathsf{PRE}}_{\mathsf{Citizen}}$. Hence, $\mathsf{IDL}^* = ((A_1, c_{a_1}), \ldots, (A_k, c_{a_k}))$ is a sequence containing the encrypted attributes and additionally the encrypted ssPINs.

5. Based on this IDL^*, a BDS template T is generated, which defines the value pairs (A_i, c_{a_i}) to be redactable by the citizen. This template is then signed by the SRA using *BDS.S*. The BDS template signing process outputs the template signature σ_{T} and the template dependent private key $\mathsf{sk}^{\mathsf{BDS,T}}_{\mathsf{Citizen}}$ for the Citizen (i.e., only the citizen holding this key is able to redact data and to create signed message instances).

6. In the last registration step, the following data are stored on the corresponding citizen card: the BDS Template T representing the IDL^*, the Template signature σ_{T}, the Template dependent private key $\mathsf{sk}^{\mathsf{BDS,T}}_{\mathsf{Citizen}}$, and the PRE decryption key of the Citizen $\mathsf{sk}^{\mathsf{PRE}}_{\mathsf{Citizen}}$.

[9] The SRA represents the originator and the citizen the proxy in terms of BDS.

The PRE encryption key of the Citizen ($\mathsf{pk}^{\mathsf{PRE}}_{\mathsf{Citizen}}$), the public parameters for BDS ($\mathsf{pp}^{\mathsf{BDS}}$) and PRE ($\mathsf{pp}^{\mathsf{PRE}}$) are stored elsewhere, for instance in the appropriate certificates or public repositories.

Identification and Authentication Process: Similar to the registration, the identification and authentication process is designed to require minimal changes to the existing infrastructure. The main changes affect the (centrally deployed) identity provider MOA-ID and the client middleware. The slight changes on the service provider's side concerns the provisioning of its proxy re-encryption key and the decryption of the received identity attributes. Both issues can be realized quite straightforwardly. The sequence diagram of the identification and authentication process is illustrated in Figure 4. The entire procedure consists of the following steps:

1. The Citizen wants to access an application deployed and running at a service provider.

2. The service provider redirects the Citizen to MOA-ID to request authentication. The authentication request holds the information in which sector the service provider operates and the PRE encryption key $\mathsf{pk}^{\mathsf{PRE}}_{\mathsf{SP}}$ of the service provider.

3. MOA-ID sends a request to the Citizen to get the Citizen's identity data and signature.

4. The Citizen reads the BDS template, holding IDL^*.

5. Due to data protection regulations, following redactions must be made: The sourcePIN and all pre-generated ssPINs not representing the given sector must be redacted out of IDL^*; i.e., only the corresponding sector (in encrypted form) stays visible. In addition to these legally required redactions, the Citizen is able to redact more identity attributes out of IDL^*, which the citizen does not want to be sent to the service provider. For instance, the Citizen may redact the name, but the date of birth is still available (as encrypted data).

6. The BDS instance (message) M is generated. This message includes the redacted IDL^* and additional information:

   - Current date and time
   - Application data (e.g., application name, country in which the application is deployed, etc.)
   - Technical parameters (e.g., URL of the application, corresponding sector of the application, etc.)

   This message is instantiated and signed by the Citizen using the private key $\mathsf{sk}^{\mathsf{DSS}}_{\mathsf{Citizen}}$ and the template dependent private key $\mathsf{sk}^{\mathsf{BDS,T}}_{\mathsf{Citizen}}$. This outputs the message signature σ_{M}.

7. The Citizen generates a re-encryption key $\mathsf{rk}^{\mathsf{PRE}}_{\mathsf{Citizen}\to\mathsf{SP}}$ for MOA-ID based upon the PRE encryption key of the service provider $\mathsf{pk}^{\mathsf{PRE}}_{\mathsf{SP}}$ and the Citizen's PRE decryption key $\mathsf{sk}^{\mathsf{PRE}}_{\mathsf{Citizen}}$ by running $PRE.RK$.

8. The Citizen returns the identity data, consisting of the BDS message signature σ_{M} and the BDS message M, to MOA-ID. In addition, this response includes the re-encryption key $\mathsf{rk}^{\mathsf{PRE}}_{\mathsf{Citizen}\to\mathsf{SP}}$, the BDS public parameter $\mathsf{pp}^{\mathsf{BDS}}$, and the PRE public parameter $\mathsf{pp}^{\mathsf{PRE}}$ (the latter is optional, since the parameters are assumed to be public).

9. MOA-ID verifies the BDS message signature. In case this verification is positive, the message is authentic and a valid instance of the BDS template as defined by the SRA.

10. MOA-ID performs the re-encryption of the identity attributes in the BDS message M, thus creating a message M^* representing the disclosed encrypted identity attributes $(\mathsf{rc}_{\mathsf{a_i}})^{\mathsf{m}}_{\mathsf{i=1}}$ re-encrypted for the service provider.

11. MOA-ID is obliged to delete the re-encryption key, due to security considerations[10].

12. MOA-ID creates an assertion Assert holding the available re-encrypted identity attributes and signs it using its private key $\mathsf{sk}^{\mathsf{DSS}}_{\mathsf{MOA-ID}}$.

13. MOA-ID transmits this signed assertion (Assert and σ_{A}) to the service provider.

14. The service provider verifies the assertion signature and proceeds if the assertion is valid.

15. The service provider decrypts the attributes in M^* using its PRE decryption key $\mathsf{sk}^{\mathsf{PRE}}_{\mathsf{SP}}$. Thus, only these identity attributes are revealed for the service provider.

16. Depending on the available (decrypted) identity data of the Citizen and the corresponding access rights, the service provider is able to grant or deny access.

Use of Proxy Re-Encryption in Practice: In the above discussion, for the sake of simplicity of presentation, we have assumed that the attribute values are encrypted using the proxy re-encryption scheme directly. However, due to efficiency reasons, in a practical application it makes more sense to use a hybrid approach, i.e., to encrypt *each* attribute using a *distinct* random key of a secure symmetric encryption scheme and then encrypt the respective key using the proxy re-encryption key.

5. DISCUSSION AND CONCLUSIONS

In this section we discuss the different approaches for (selective) disclosure comparing it to our new model. Basis for our discussion are the following selected criteria (based upon the requirements given in Section 3.2):

Privacy and Selective Disclosure: How much (identity) data are revealed to the identity provider and service provider? Is the approach applicable as an Identity as a Service model?

[10] In case the service provider gets compromised and MOA-ID is still holding the re-encryption key, MOA-ID and the service provider might cooperate. This might enable MOA-ID being able to re-encrypt the attributes for the forged service provider.

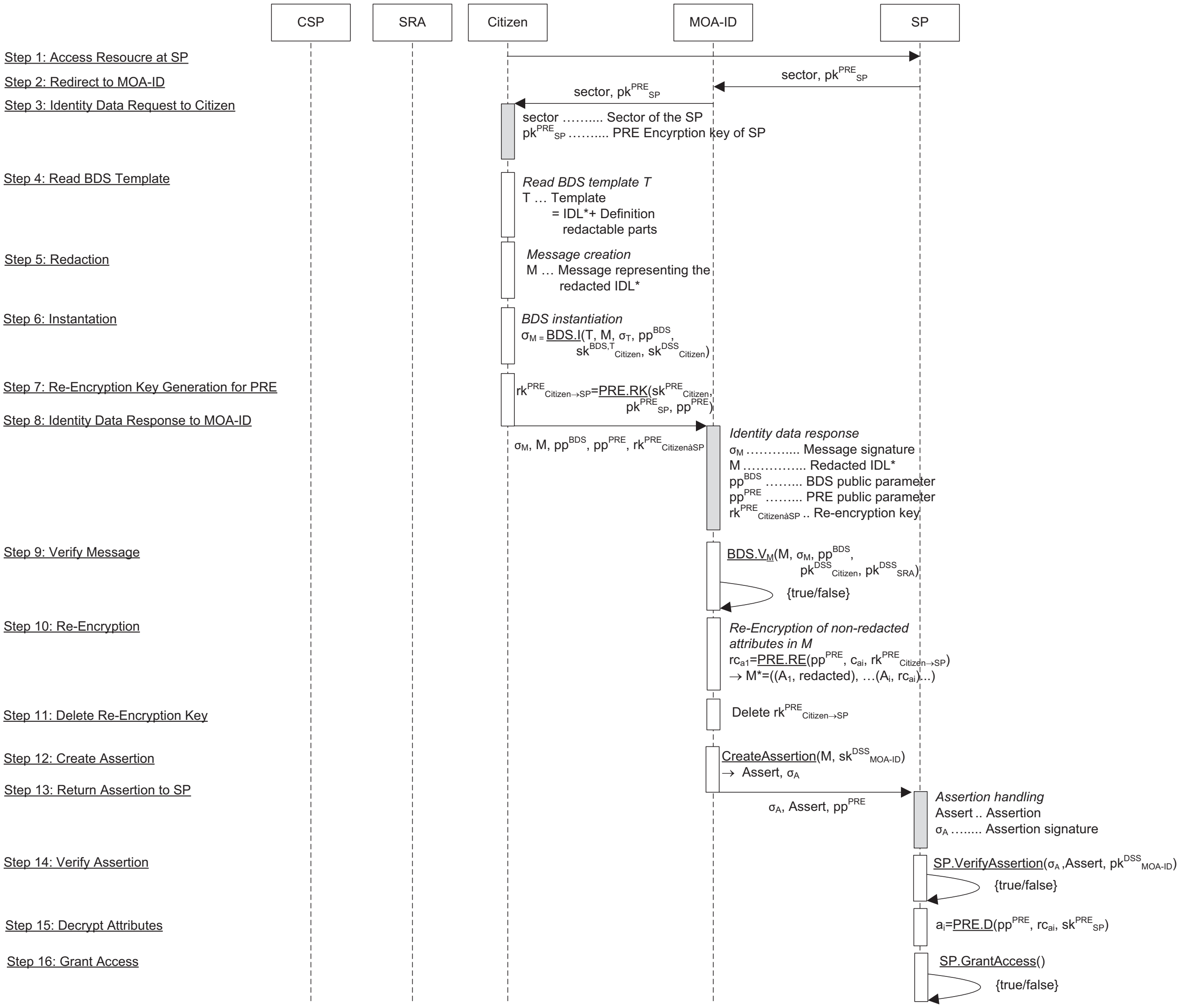

CSP
SRA
Citizen
MOA-ID
SP
Step 1: Access Resoucre at SP
Step 2: Redirect to MOA-ID
sector, pk^{PRE}_{SP}
sector, pk^{PRE}_{SP}
Step 3: Identity Data Request to Citizen
sector ……... Sector of the SP
pk^{PRE}_{SP} ……... PRE Encyrption key of SP
Step 4: Read BDS Template
Read BDS template T
T … Template
= IDL*+ Definition redactable parts
Step 5: Redaction
Message creation
M … Message representing the redacted IDL*
Step 6: Instantation
BDS instantiation
σ_M = BDS.I(T, M, σ_T, pp^{BDS}, $sk^{BDS,T}_{Citizen}$, $sk^{DSS}_{Citizen}$)
Step 7: Re-Encryption Key Generation for PRE
$rk^{PRE}_{Citizen\rightarrow SP}$=PRE.RK($sk^{PRE}_{Citizen}$, pk^{PRE}_{SP}, pp^{PRE})
Step 8: Identity Data Response to MOA-ID
σ_M, M, pp^{BDS}, pp^{PRE}, $rk^{PRE}_{CitizenàSP}$
Identity data response
σ_M ………... Message signature
M …………... Redacted IDL*
pp^{BDS} ……... BDS public parameter
pp^{PRE} ……... PRE public parameter
$rk^{PRE}_{CitizenàSP}$ .. Re-encryption key
Step 9: Verify Message
$BDS.V_M$(M, σ_M, pp^{BDS}, $pk^{DSS}_{Citizen}$, pk^{DSS}_{SRA})
{true/false}
Step 10: Re-Encryption
Re-Encryption of non-redacted attributes in M
rc_{a1}=PRE.RE(pp^{PRE}, c_{ai}, $rk^{PRE}_{Citizen\rightarrow SP}$)
→ M*=((A_1, redacted), …(A_i, rc_{ai})…)
Step 11: Delete Re-Encryption Key
Delete $rk^{PRE}_{Citizen\rightarrow SP}$
Step 12: Create Assertion
CreateAssertion(M, sk^{DSS}_{MOA-ID})
→ Assert, σ_A
Step 13: Return Assertion to SP
σ_A, Assert, pp^{PRE}
Assertion handling
Assert .. Assertion
σ_A ……. Assertion signature
Step 14: Verify Assertion
SP.VerifyAssertion(σ_A ,Assert, pk^{DSS}_{MOA-ID})
{true/false}
Step 15: Decrypt Attributes
a_i=PRE.D(pp^{PRE}, rc_{ai}, sk^{PRE}_{SP})
Step 16: Grant Access
SP.GrantAccess()
{true/false}

Figure 4: Sequence diagram of identification and authentication processes.

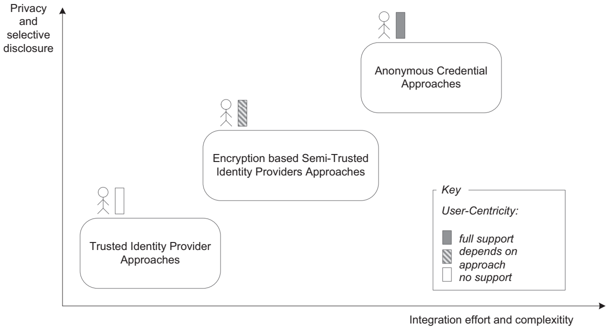

Privacy and selective disclosure
Anonymous Credential Approaches
Encryption based Semi-Trusted Identity Providers Approaches
Trusted Identity Provider Approaches
Key
User-Centricity:
full support
depends on approach
no support
Integration effort and complexity

Figure 5: Overview assessment.

User-Centricity: Is the user herself able to disclose only parts of her identity data towards the service provider and the identity provider?

Integration effort and complexity: How much effort is required by adopting the approach? How complex is the integration of the approach, especially for the service provider?

Figure 5 gives a brief assessment of the different approaches discussed in Section 2 concerning these criteria. The trusted identity provider approaches represent the state-of-the-art and are already deployed in many cases. Thus, the integration effort and complexity are quite low. Due to requiring full trust on the identity provider, these approaches do not achieve a high grade on privacy, user-centricity, and selective disclosure. In contrast, anonymous credential systems allow a privacy-aware and user-centric (selective) disclosure of identity data. They achieve both anonymity and unlinkability with respect to the service provider and identity provider. Nevertheless, these systems require complex operations and additional infrastructure and lack on deployment in practice. Semi-trusted identity provider approaches using proxy re-encryption steer a middle course. While they achieve anonymity of the citizen regarding the identity provider because of the encrypted attributes, they are not able to avoid linkability (identity providers, however, can only link actions to pseudonyms). Compared to the trusted identity providers approach, these approaches increase (slightly) the integration effort and complexity, but seem far more practical than anonymous credentials.

The approach of Nuñez et al. [19] uses proxy re-encryption and is therefore advantageous when the identity provider is not fully trusted in terms of privacy. Nevertheless, drawbacks are that this approach is server-side, and thus the user still needs to trust the identity provider that only the amount of data desired by the user is actually transferred to the service provider. Hence, the approach cannot be seen as user-centric in terms of selective disclosure. Moreover, the encrypted attributes stored on the OpenID provider may be self-asserted, hence no qualified and authentic attributes such as required when using eIDs can be provided to the service provider.

The second proxy re-encryption-based approach for semi-trusted identity providers of Zwattendorfer and Slamanig [25] allows a selective disclosure of identity data. Apart the fact that their approach is strongly tailored to the Austrian eID system, it is not fully user-centric as the user data are encrypted for a trusted third party and not the user herself. Additionally, the approach requires a pre-registration of identity providers and service providers at a registration authority, which issues the identity data. This strikingly decreases the practicability and extensibility of this approach.

In contrast, in our proposed model the user gets full control about her data. On the one hand, the identity data are encrypted for the user and the user herself generates the proxy re-encryption key (defining which service provider is able to decrypt the identity data). On the other hand, the user is able to define which data are disclosed before it is sent to the identity provider. Thus, the privacy and user-centric selective disclosure requirement is fulfilled. Furthermore, our approach does not need any pre-registration of the identity provider or the service provider. Finally, additional infrastructure components except additional decryption functionality at the service provider are not needed. Hence, our approach allows for a convenient integration into existing infrastructures and protocols.

Acknowledgements

We thank the anonymous reviewers for their valuable feedback. The first author has been partly supported by the European Commission through project FP7-FutureID, grant agreement number 318424.

6. REFERENCES

[1] G. Ateniese, K. Fu, M. Green, and S. Hohenberger. Improved proxy re-encryption schemes with applications to secure distributed storage. *ACM Trans. Inf. Syst. Secur.*, 9(1):1–30, 2006.

[2] M. Bauer, M. Meints, and M. Hansen. D3.1: Structured Overview on Prototypes and Concepts of Identity Management System. FIDIS, 2005.

[3] P. Bichsel, J. Camenisch, T. Groß, and V. Shoup. Anonymous credentials on a standard java card. In *ACM CCS*, pages 600–610. ACM, 2009.

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

[5] C. Brzuska, H. C. Pöhls, and K. Samelin. Non-Interactive Public Accountability for Sanitizable Signatures. In *EuroPKI*, volume 7868 of *LNCS*. Springer, 2012.

[6] J. Camenisch and A. Lysyanskaya. An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation. In *Advances in Cryptology - EUROCRYPT 2001*, volume 2045 of *LNCS*, pages 93–118. Springer, 2001.

[7] European Commission. IDABC. 2009. eID Interoperability for PEGS: Update of Country Profiles., 2009.

[8] European Parliament and the Council. 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 . `http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:en:HTML`, 1995.

[9] European Union. Directive 1999/93/EC of the European Parliament and of the Council of 13. December 1999 on a community framework for electronic signatures.

[10] C. Hanser and D. Slamanig. Blank Digital Signatures. In *ACM ASIACCS '13*, pages 95–106. ACM, 2013. `http://eprint.iacr.org/2013/130`.

[11] R. Johnson, D. Molnar, D. X. Song, and D. Wagner. Homomorphic Signature Schemes. In *CT-RSA '02*, volume 2271 of *LNCS*, pages 244–262. Springer, 2002.

[12] A. Jøsang and S. Pope. User centric identity management. *AusCERT 2005*, 2005.

[13] A. Jøsang, M. A. Zomai, and S. Suriadi. Usability and privacy in identity management architectures. In *ACSW '07*, pages 143–152, 2007.

[14] H. Leitold, A. Hollosi, and R. Posch. Security Architecture of the Austrian Citizen Card Concept. In *ACSAC 2002*, pages 391–402, 2002.

[15] H. Leitold and B. Zwattendorfer. STORK: Architecture, Implementation and Pilots. In *ISSE 2010*, pages 131–142, 2010.

[16] M. Margraf. The new german id card. In *ISSE 2010 Securing Electronic Business Processes*, pages 367–373. Vieweg+Teubner, 2011.

[17] Modinis. The Status of Identity Management in European eGovernment initiatives. Deliverable D3.5, 2006.

[18] I. Naumann and G. Hogben. Privacy Features of European eID Card Specifications. Technical report, European Network and Information Security Agency (ENISA), 2009.

[19] D. Nuñez, I. Agudo, and J. Lopez. Integrating OpenID with Proxy Re-Encryption to enhance privacy in cloud-based identity services. In *IEEE CloudCom 2012*, pages 241 – 248, 2012.

[20] J. Palfrey and U. Gasser. CASE STUDY: Digital Identity Interoperability and eInnovation. Berkman Publication Series,, 2007.

[21] Republic of Austria. *Austrian Federal Act on Provisions facilitating electronic communications with public Bodies; part I, Nr. 10/2004*. Federal law Gazette, 2004.

[22] A. Sabouri, I. Krontiris, and K. Rannenberg. Attribute-Based Credentials for Trust (ABC4Trust). In *TrustBus 2012*, volume 7449 of *LNCS*, pages 218–219. Springer, 2012.

[23] A. Siddhartha. National e-id card schemes: A european overview. *Inf. Secur. Tech. Rep.*, 13(2):46–53, May 2008.

[24] R. Steinfeld, L. Bull, and Y. Zheng. Content Extraction Signatures. In *ICISC 2001*, volume 2288 of *LNCS*, pages 285–304. Springer, 2001.

[25] B. Zwattendorfer and D. Slamanig. On Privacy-Preserving Ways to Porting the Austrian eID System to the Public Cloud. In *SEC 2013*, AICT, pages 300–314. Springer, 2013.

[26] B. Zwattendorfer and D. Slamanig. Privacy-preserving realization of the stork framework in the public cloud. In *SECRYPT*, pages 419–426, 2013.

APPENDIX

A. CRYPTOGRAPHIC BUILDING BLOCKS

In the following we give a more formal description of the cryptographic primitives used in our model.

Proxy Re-Encryption.

A proxy re-encryption scheme (PRE) is a tuple (S, K, RK, E, RE, D) of polynomial-time algorithms. The algorithm $PRE.S$ represents the setup and produces system parameters $\mathsf{pp}^{\mathsf{PRE}}$. Every algorithm has access to these parameters. $PRE.K$ is a probabilistic key generation algorithm that takes a security parameter κ and outputs a private and public key pair $(\mathsf{sk}_{\mathsf{A}}^{\mathsf{PRE}}, \mathsf{pk}_{\mathsf{A}}^{\mathsf{PRE}})$ for party A. The re-encryption key generation algorithm $PRE.RK$ takes as input a private key $\mathsf{sk}_{\mathsf{A}}^{\mathsf{PRE}}$ and a different public key $\mathsf{pk}_{\mathsf{B}}^{\mathsf{PRE}}$, and outputs a re-encryption key $\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}$. The probabilistic encryption algorithm $PRE.E$ gets a public key $\mathsf{pk}_{\mathsf{A}}^{\mathsf{PRE}}$ and a plaintext M and outputs $c_i = PRE.E(\mathsf{pk}_{\mathsf{A}}^{\mathsf{PRE}}, M)$. The (probabilistic) re-encryption algorithm gets as input a ciphertext c_A under $\mathsf{pk}_{\mathsf{A}}^{\mathsf{PRE}}$ and a re-encryption key $\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}$ and outputs a re-encrypted ciphertext $c_B = PRE.RE(\mathsf{rk}_{\mathsf{A}\to\mathsf{B}}^{\mathsf{PRE}}, c_A)$ under $\mathsf{pk}_{\mathsf{B}}^{\mathsf{PRE}}$. The decryption algorithm $PRE.D$ takes the private key $\mathsf{sk}_{\mathsf{B}}^{\mathsf{PRE}}$ and a ciphertext c_B and outputs $M = PRE.D(\mathsf{sk}_{\mathsf{B}}^{\mathsf{PRE}}, c_B)$ or an error $\perp$.

Digital Signatures.

A digital signature scheme (DSS) is a triple (K, S, V) of poly-time algorithms, whereas $DSS.K$ is a probabilistic key generation algorithm that takes a security parameter κ and outputs a private and public key pair $(\mathsf{sk}^{\mathsf{DSS}}, \mathsf{pk}^{\mathsf{DSS}})$. The probabilistic signing algorithm $DSS.S$ takes as input a message $M \in \{0,1\}^*$ and a private key $\mathsf{sk}^{\mathsf{DSS}}$, and outputs a signature σ. The verification algorithm $DSS.V$ takes as input a signature σ, a message $M \in \{0,1\}^*$ and a public key $\mathsf{pk}^{\mathsf{DSS}}$, and outputs a single bit $b \in \{\mathtt{true}, \mathtt{false}\}$ indicating whether σ is a valid signature for M. We note that in practice one typically employs the hash-then-sign paradigm, i.e., instead of inputting M into $DSS.S$ and $DSS.V$, one inputs $H(M)$ where H is a suitable cryptographic hash function.

Redactable Signatures.

A redactable signature scheme (RS) is a tuple (K, S, V, R) of polynomial-time algorithms. The algorithm $RS.K$ gets a security parameter κ and generates a private and public key pair $(\mathsf{sk}^{\mathsf{RS}}, \mathsf{pk}^{\mathsf{RS}})$. The signing algorithm $RS.S$ gets as input the signing key $\mathsf{sk}^{\mathsf{RS}}$ and a message $m = (m[1], \ldots, m[\ell])$, $m[i] \in \{0,1\}^*$ and outputs a signature $\sigma = RS.S(sk^{RS}, m)$. The verification algorithm $RS.V$ gets as input a public key $\mathsf{pk}^{\mathsf{RS}}$, a message $m = (m[1], \ldots, m[\ell])$, $m[i] \in \{0,1\}^*$, a signature σ, and outputs a single bit $b = RS.V(pk^{RS}, m, \sigma)$, $b \in \{\mathtt{true}, \mathtt{false}\}$, indicating whether σ is a valid signature for m. The redaction algorithm $RS.R$ takes as input a message $m = (m[1], \ldots, m[\ell])$, $m[i] \in \{0,1\}^*$, a public key $\mathsf{pk}^{\mathsf{RS}}$, a signature σ, and a list MOD of indizes of blocks to be redacted. It returns a modified message with redacted blocks and a signature pair $(\hat{m}, \hat{\sigma}) = RS.R(m, pk^{RS}, \sigma, \mathsf{MOD})$ or an error $\perp$. In case that it should be publicly verifiable that the modified message $\hat{m}$ has been produced by some particular redactor (non-interactive public accountability [5]), the redactor is required to create a conventional digital signature on it.

Blank Digital Signatures.

A blank digital signature scheme (BDS) is defined as a tuple (K, S, V_T, I, V_M) of polynomial-time algorithms. Algorithm $BDS.K$ on input of a security parameter κ and an upper bound for the template size t generates public parameters $\mathsf{pp}^{\mathsf{BDS}}$. $BDS.S$ takes a template T, the public parameters $\mathsf{pp}^{\mathsf{BDS}}$, the private signing key of the originator $(\mathsf{sk}_{\mathsf{O}}^{\mathsf{DSS}})$ and the public signing key of the proxy $(\mathsf{pk}_{\mathsf{P}}^{\mathsf{DSS}})$, and outputs a template signature σ_{T} and a private template signing key for the proxy $\mathsf{sk}_{\mathsf{P}}^{\mathsf{BDS,T}}$. Algorithm $BDS.V_T$ given the template T, the template signature σ_{T}, the public parameters $\mathsf{pp}^{\mathsf{BDS}}$, the public signing keys of originator and proxy $(\mathsf{pk}_{\mathsf{O}}^{\mathsf{DSS}}, \mathsf{pk}_{\mathsf{P}}^{\mathsf{DSS}})$, and the template signing key of the proxy $\mathsf{sk}_{\mathsf{P}}^{\mathsf{BDS,T}}$, checks whether σ_{T} is a valid signature for T or not. $BDS.I$ on input of a template T, a corresponding instance M, a signature on the template σ_T, as well as the public parameters $\mathsf{pp}^{\mathsf{BDS}}$, the private template signing key $\mathsf{sk}_{\mathsf{P}}^{\mathsf{BDS,T}}$, and the private signing key of the proxy $\mathsf{sk}_{\mathsf{P}}^{\mathsf{DSS}}$ outputs a signature on the message σ_{M}. The algorithm $BDS.V_M$, when given an instance M of a template T, a signature on this instance σ_{M}, the public system parameters $\mathsf{pp}^{\mathsf{BDS}}$, and the public signing keys of proxy and originator $(\mathsf{pk}_{\mathsf{O}}^{\mathsf{DSS}}, \mathsf{pk}_{\mathsf{P}}^{\mathsf{DSS}})$, verifies whether σ_{M} is a valid signature on M and if M is a correct instantiation of T.