A Privacy-Enhanced User-Centric Identity and Access Management Based on Notary

Hendri Nogueira, Rick Lopes de Souza and Ricardo Felipe Custódio
Laboratory of Computer Security
Federal University of Santa Catarina
Florianópolis-SC, Brazil
Email: {jimi, rick.lopes, custodio}@inf.ufsc.br

***Abstract*—Identity and Access Management (IAM) systems aim to control of users' attributes for authentication, authorization and accountability processes. Public Key Certificates (PKCs), like the X.509 standard, use asymmetric key pairs to support digital signatures, authentication processes and to increase the trust in the communication. Nevertheless, the PKC does not concern itself with the management of users' attributes and their privacy to be used as an IAM system. We present a privacy-enhanced identity and access management architecture, addressing the user's management of his attributes and the privacy. With the user-centric paradigm and through the use of Identity-Based Cryptography (IBC), the model architecture is composed by a user-centric public key infrastructure. The asymmetric key pair enables the user to determine the control and the anonymity of his own attributes and the Notarial Authority validates the attributes claimed by the user. Our model aims for total control for the user in authentication and authorization procedures. Users can decide which attributes they want to disclose and which identity to use (e.g., real identity, pseudonym, anonymity).**

***Keywords*—User-centric; Identity Management; Notary; Attributes; Privacy-Enhancing; IBC.**

I. INTRODUCTION

Many standards related to Authentication and Authorization (AA) processes demand user's registration in the Services Providers (SPs), storing their attributes in the SPs database, and consequently replicating the attributes without users' control. Others divide responsibilities by those which manages uses' attributes and authenticate them, called the Identities Providers (IdPs), and those which only provide services for authorized users. Independently of that, the users' attributes need to be managed in a safe way and can not be used for other purposes than what was determined. Additionally, the AA systems must concern about the users' privacy and provide secure mechanisms to protect the users' identity and their related attributes.

The use of asymmetric cryptography keys have advantages in binding a key pair with the subject's attributes. The X.509 PKC, for example, can be applied to automated identification, authentication, digital signatures, access control and authorization functions in a digital environment [1], [2]. However, PKC is not recommended to be used for authorization procedures and when the user's privacy is a necessity. Additionally, the management of the PKCs by an X.509 Public Key Infrastructure (PKI) is difficult and expensive, requiring a lot of effort for its management and maintenance and leaving doubts as to the cost-benefit as regards its functionality [3]–[5].

The amount of verification procedure of a certificate and the revocation mechanism might be a disadvantage in some environments and situations with limited resources [6]. If the PKI is composed by many certification authorities and generating a large certification path until the end user certificate, the PKC's verification may not always be performed quickly. If the PKC's revocation constantly happens before the end of its validity, it interferes in the issuance of the certificates' revocation states in real-time.

Since most attributes for access control, role, and permission do not have a long lifetime (i.e., more than a certificate valid period), it is not recommended to include these types of attributes into a PKC. Moreover, an end user certification authority may not be the responsible for the management of those user's attributes, what means that the user's attributes values could be questionable. In this case, X.509 Attribute Certificates (ACs) could be a solution [7]. To provide a better security, PKCs and ACs should work together, but two different infrastructures are necessary to manage each one, PKI and X.509 Privilege Management Infrastructure (PMI) respectively [8]. However, this inherits the same issues from PKI and would increase the complexity, the costs, the human and computational resources.

The management of users' attributes in a PKC and how they are accessed do not concern about the user's privacy. As a PKC can be used as an off-line token to AA procedures, it needs to provide a sufficient amount of attributes to support the user in different situation. To reduce the amount of data in a unique PKC, an alternative could be the use of many PKCs with different attributes, but it is costly for the user. Furthermore, when a user provides his PKC, the verification system reads all the information in the certificate, even though what is not necessary for that procedure. Other privacy issue is related to the user's identity, which one (or more) identification attribute is used to bind with the user public key. Every time that the user uses the same PKC, the identification attribute and the public key bind to the action realized, and this can be traceable.

a) Contribution: Beyond the problems we have stated above, we present the concept of user-centric PKI architecture for identity and access management to improve the management, the disclosure, the users' control on their attributes and their privacy. The model explores two PKI problems: (1) the high costs of a PKC for end-users and leaving doubts as to the cost-benefit as regards for identity and access management, and (2) the X.509 PKC privacy deficiency, allowing the real iden-

 ISBN: 978-1-61208-305-6

tification of users, forcing users to reveal more attributes than needed, and enabling users' on-line transactions linkable across different websites. With this intention, our model introduce a new way that a user gets and uses an asymmetric cryptography key pair to claim his attributes to service providers and been validated by the Notary. Though the use of the identity-based cryptography and the user-centric paradigm, the user issues and manages their own private keys and issues self-signed assertions.

b) Outline: We start this paper by describing the related works. We also describe about privacy in IAM systems and we introduce about identity-based cryptography (with each correlated works), in Section III and Section IV, respectively. Next, we present our proposal followed by definitions (Section V). More practical descriptions of our idea, including a description of the procedures that players in our scheme perform are also shown (Section V-B). Afterwards, we describe some analysis about our model (Section VI). Finally, we present our considerations and future works (Section VII).

II. Related Work

Facing of the X.509 PKI problems described in the Introduction section, several works treat or propose alternatives to the revocation's problems. The Hormann et al.'s work, for example, aims at improving the existing revocation mechanism [9], while Scheibelhofer proposes a PKI without revocation checking and reducing the verification processes [10]. Faced with various revocation mechanisms, both existing and proposed, Ofigsbo et al. analyzed the cost of some mechanisms [11].

Alternative PKI models and concepts were created to give a different architecture of a PKI also. Focusing on digital signature issues, Moecke et al. [12] proposed a change to the form in which certificates are issued. Some optimizations were proposed, as in Vigil et al.'s [13] work. Vigil et al. [14] also proposed a new approach to X.509 PKI, based on notarials' responsibilities to support long-term signatures on documents. On the other hand, those works do not focus on users' attributes management and privacy.

The use of the notary responsibility is not new. Adams and Zuccherato's work [15] described a general notary service and protocols that notarial authorities validate signatures and provide up-to-date information regarding the status of certificates. This is also usable to extend the lifetime of a signature beyond key expiry or revocation. Another work based on notary is the Chao-yang's work that improved some computer notary system protocols to decrease the replay attack vulnerability in the agreement communication [16].

Another PKI scheme, the Simple Public Key Infrastructure (SPKI) [17] proposes and simplifies the PKI architecture and focuses on authorization processes, binding one key with a user's authorization [18]. Additionally, the Simple Distributed Security Infrastructure (SDSI) combines the SPKI design and the definition of groups to issue certificates to group membership [19]. SPKI/SDSI is limited because there is no formal bondage of trust between entities involved and a member can make an inquiry on behalf of its group.

III. Privacy in Identity and Access Management Systems

Concerning the existence of different specifications and frameworks for IAM systems, Jøsang and Pope's work [20] reports differences between the paradigms available. They concluded that the user-centric paradigm improves the user experience and the security of on-line service provision as a whole. Moreover, the user-centric paradigm aims the user's control at the different aspects of his identity, which it is used in different contexts and situations (called "partial identities"), and enhancing his privacy.

In on-line systems, where identities providers create access tokens on demand (e.g., SAML [21], OpenID [22], WS-Federation [23]) and also supporting a Single Sign-On (SSO) mechanism [24], they can lead to the impersonation of their users and the tracking of users' actions on-line. Systems with off-line token creation, such as X.509 certificates and some WS-Trust profiles [25] force the user to reveal more attributes than needed (as otherwise the issuer's signature cannot be verified) and make the on-line transactions linkable across different websites.

The privacy is made of terminologies, e.g., pseudonymity, anonymity, linkability, detectability, observability, and they provide different levels of privacy [26]. To point out two terminology above, anonymity and accountability are the extremes points related to the user linkability. Pseudonymity comprises all subset between and including the extremes above and all degrees of linkability. In each specific case, the strength of anonymity depends on the knowledge of certain parties about the linking relative to the chosen attacker model.

Privacy-enhancing identity management systems make the flow of personal data transparent and give users the control of their individual digital identity, i.e., their individual partial identities in an on-line world. The European PRIME project (Privacy and Identity Management for Europe) allows users to control the disclosure of their personal information and allows users to authenticate with anonymous credentials [27]. The PRIME architecture requires service providers to change their infrastructure server and the user needs to install the client side. The PRIME project succeeded to the PrimeLife (Privacy and Identity Management for Europe for Life) project. PrimeLife implemented the PrimeLife Policy Language [28].

Other anonymous credential system are the Idemix (short for "identity mix") [29] and the U-Prove [30]. The Idemix enables authentication, privacy and guarantees "anonymity" on the Internet. Nevertheless, the Idemix architecture is such complex and costly to implement for the issuer. The U-Prove specification uses cryptographic mechanisms which trusted parties issue "tokens" to users that contains user's attributes. The user is enable to select which attributes he wants to disclose from his "token" and the authentication could be in a anonymous way. However, U-Prove specification provides a revocation mechanism for the users' credentials by blacklisting the token identifier in which this turns the tokens linkable.

IV. Identity-Based Cryptography

Proposed by Shamir, the Identity-Based Cryptography (IBC) concept is based on the use of a string as a public key

 ISBN: 978-1-61208-305-6

for encryption and signature procedures [31]. The string is the user identity information (e.g., an email, name, IP). As a result, IBC significantly reduces the system complexity and the cost for establishing and managing the public key in a PKI [32], [33]. In 2001, Boneh and Franklin [34] and Cocks [35] solved the Shamir's identity-based encryption open problem. Baek et al. surveyed the state of research on identity-based cryptography [36].

Another approach is the Attribute-Based Signatures (ABS) that allows a party to sign a message with fine-grained control over identifying information. ABS is based on identity-based encryption in which each identity is considered as a set of descriptive attributes [37]. There are works use attribute-based signatures and attribute-based encryption to develop a cryptosystem for fine-grained sharing of encrypted data [38] and to propose a threshold attribute-based signatures (t-ABS) [39]. In our proposal, we use the concepts of IBC to compose our architecture's model and archive our goal.

In an IBC scheme, a trusted third party called Private Key Generator (PKG), aka Key Generation Center, is a trust authority responsible for generating the user's secret key. To be able to issue secret keys, the PKG needs to create a master secret key (*msk*) and the correspondent master public key (the public parameters and the public key itself) – *mpk*. The PKG's *mpk* is widely distributed and any party can compute a public key corresponding to an identity (*id*) by combining the master public key with the identity value. To get the corresponding secret key, it is necessary to authenticate through the PKG with that *id*. Then the PKG uses its master secret key and the user's *id* value to issue the corresponding secret key.

Some of the IBC advantages related to a standard PKI are: the public keys are derived from identifiers and than eliminates the need for a public key distribution infrastructure; the authenticity of the public keys is guaranteed implicitly as long as the transport of the secret keys to the corresponding user is kept secure; a compromised end-user secret key only exposes messages encrypted/signed with that particular *id* used to compute the secret key; no CRLs are needed; it is certificateless.

On the other hand, IBC also has disadvantages. Some of them are: a PKG needs to maintain a authentication infrastructure; the private key extraction has a very high exposure to man-of-the-middle attack; the PKGs do not interact with each other; it is necessary to support revocation of *ids* and consequently a well-defined expiry date for secret keys; and there is inherent key escrow, i.e., the users' secret key is known to the PKG.

V. User-Centric Public Key Infrastructure Based on Notaries

The User-Centric Public Key Infrastructure based on notaries (UCPKI) focuses on the management of users' attributes, where the user has more control and privacy over the disclose of his attributes to the services providers. UCPKI also addresses privacy-enhancement to the management of identity and access architecture, enabling anonymity, unlinkability and making the user untraceable. Based on the real world of notary responsibilities and services, the model's architecture has Notarial Authorities (NAs) that are trusted third parties responsible for verifying users' attributes as well as validating them. The model's architecture adopts the concept of identity-based cryptography and the user-centric paradigm in which the users issue and manage their own secret keys.

Considering that our model is user-centric and the IBC architecture needs a trust authority to issues secret keys based on identifications thus, it is the user who is going to realize that role, i.e., the role of a private key generator. As a consequence, the user maintains control of his identities used in each communication.

A. Components

In this subsection, we define the concepts involved in our model. We define two main entities: Attribute Registration Authority (ARA) and Notarial Authority (NA). UCPKI uses a Trust-service Status List (TSL) to keep the management of the trusted ARAs and to know the relation of each NA with the ARAs. Support to enhance the users' privacy is given by IBC.

1) Attribute Registration Authority: An Attribute Registration Authority is an entity responsible for registering attributes for the user (e.g., name, surname, e-mail address, occupation), storing the information in its trusted database system, and keeping attributes up to date. An ARA has to be responsible for, at least, one attribute from the user. Each ARA has an asymmetric cryptographic key pair to be used in the communication's workflow. The ARA's information and its public key are managed by a Trust-service Status List. Some examples of an ARA are the entities responsible for registering users' attributes for governmental, professional, or even business purposes.

2) Notarial Authority: A Notarial Authority is a point of trust responsible for receiving self-signed assertions from users and validating users' attributes. The NA communicates with the attribute registration authorities to confirm the correctness of the user's attributes. The validation of the assertion results in the assertion's signature by the NA (a co-signature). This procedure certifies the truthfulness of the user's attributes. To be defined as a trust authority, each NA has an asymmetric cryptographic key pair used to sign the assertions and to make the communication secure. The trust of the public keys tied to each NA and ARA is managed by a Trust-service Status List.

3) Trust-service Status List: A Trust-service Status List (TSL) is used to manage and inform the trust between NAs and ARAs. TSL turns trustworthy information about the entities relationships, along with a historical status and the associated public keys [40]. A TSL may be composed of a list of TSLs and it is managed, signed, and published into a public trust repository by a trusted entity of its domain.

B. How it Works

First, the user needs to create a master secret key and the correspondent master public key. To keep the *msk* safe, it is created in a secure device (e.g., smartcard or USB token) and it is protected with a PIN code. After the master key pair is created, the user must register his *mpk* in each ARA's database that manages at least one attribute about him. If the ARA already has an authentication mechanism installed,

 ISBN: 978-1-61208-305-6

then the registration of the user's *mpk* can be done after the user authentication. Otherwise, the most secure way is for the registration to be done personally.

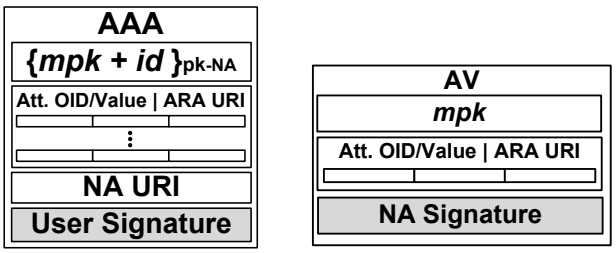
The validity of the master key pair is equally associated with the cryptographic algorithm used. If something were to happen to the user's *msk* during the time of validity, a procedure to change the registration of the user's *mpk* in the ARAs must be executed. For this change of *mpk* association in an ARA, we propose the use of a One Time Password (OTP) code [41] to facilitate the ARA's infrastructure and the user's life. In this case, the ARA does not necessarily have to maintain other authentication mechanism for the user (e.g., login and password), neither does the user need to remember his login information. The OTP code must be used only once and is given to the user after his *mpk* registration.

C. Accessing a Service Provider

To access an SP and get its resource, the user needs to choose an identity (e.g., real name, e-mail address, any string) and inform the necessaries attributes. The information is passed through a data structure, called the Attribute Authentication Assertion (AAA); see Figure 1a. Within an AAA, the user includes his (*mpk*) and his identifier ciphered with the public key of an NA ($\{mpk + id\}pk$-NA). This NA is chosen by the user preference. An AAA also contains: a set of attributes' Object Identifiers (OIDs), the attributes' values, and the referenced ARA responsible to the attributes (ARA URI); and the NA's reference (NA URI) to indicate which NA can correctly decipher the user's (*mpk* and identifier). The structure is signed by the user with the secret key corresponded to his chosen *id*.

Next, the user sends the AAA to the SP (illustrated in Figure 2 by step 1). The SP receives it and sends it (and also its public key) to the NA referenced in the AAA (step 2). The NA deciphers the user's *mpk* and identifier with its private key and uses the *mpk* with the user's *id* to verify the AAA's signature. If the signature is correct, the NA communicates to the referenced ARA to get the attributes verified (step 3). The NA sends the ARA a data structure, called Attribute Validation (AV) – see Figure 1b. An AV contains the user's *mpk* and the correspondent set of attributes' OIDs and values. Because it may have many attributes' sets related to the different ARAs, each set is verified through the correspondent *ARA URI*. All the communication is done by a secure channel to prevent the man-in-the-middle attack.

Each ARA manages the uses' attributes and the attributes are associated with the users' *mpk*. Therefore, when the ARA



(a) Attribute Authentication Assertion.

(b) Attribute Validation.

Fig. 1. Data structures used in the workflow model.

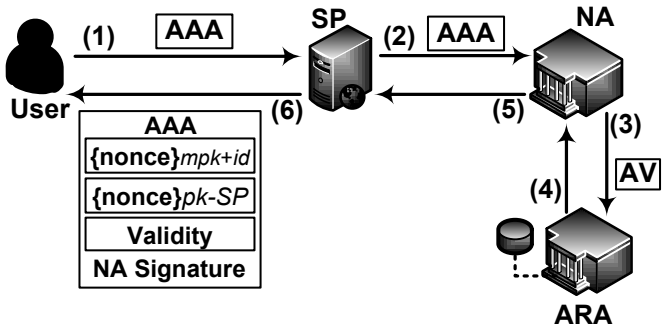


Fig. 2. Workflow to access a service provider.

receives NA's AV request, the ARA verifies the AV's signature and checks in its records if the associations of attributes' values are correct. If the ARA confirms the truth of the attributes, the ARA co-signs the AV and returns the signature as a confirmation response to the NA (step 4). After receiving all signatures from the ARAs involved, the NA generates a *nonce* to provide a challenge-response mechanism and the anonymous authentication of the user to the service provider. This *nonce* is ciphered with the user's *mpk* and the user's *id*. The NA also gets the same clear-text *nonce* and ciphers it with the SP's public key. Both nonces ciphered are attached to the AAA and then the AAA is co-signed by the NA with its private key. A validity period (e.g., a day, a week, a month) is also determined by the NA to indicate for how long those information are valid.

The co-signed AAA is sent back to the SP (step 5). The SP keeps a copy and the delivers the co-signed AAA to the user (step 6). Now, the user must authenticate (in a anonymous way) with the SP. This procedure is done by the use of the *nonce* created by the NA and included into the co-signed AAA. The user deciphers the *nonce* using the secret key related to the *id* used in the AAA. With the *nonce* in clear-text, the user ciphers again using the SP's public key and sends to the SP. The SP deciphers this cipher-text and gets the *nonce*'s value. The SP also deciphers the *nonce* included in the user's co-signed AAA and compares the two resulted values. If they were equal, the SP concludes that: the user who created the AAA is the same who has the master secret key (i.e., is the same user who created the secret key to sign the AAA with the related *id*); the attributes' values are validated through the NA; and the user is able to get the resources according to the SP's policies.

Once an AAA is co-signed by an NA, the user can reuse it with the same SP until the validity time included in the AAA. The AAA's validity could be based on the validity information included in the AAA or depending on the SP's policies. The SPs' key pair is managed by themselves and the public key is published publicly. Each NAs and ARAs' private key is managed in a secure device and the correspondent public key is managed in the TSL's domain.

VI. Analysis

The use of identity-based cryptography is essential to provide the dynamism and the facility to users in controlling which identities they want to use in each access. The IBC

procedures in our model eliminate the problems caused by the use of a public key certificate (cited in Section I) and also give the users more privacy to an identity and access management architecture. The key escrow provided by a common IBC is eliminated by the user-centric paradigm in our UCPKI model, in which the user maintains the total control of the master secret key and all secret keys related to each *id*.

The user's master secret key must be included into a secure device (e.g., token, smartcard) which the *msk* can not be moved, copied, and its usage must be protected by a password mechanism (e.g., PIN, OTP). The device should be able to realize cryptographic functions into it, like the generation of a secret key from an *id* and the signature of an AAA data structure. If the user loose his smartcard, he must do the procedure to change the registration of his *mpk* (as soon as possible) with all ARAs that manage his attributes.

The UCPKI architecture and the use of encryptions and signature procedures by the IBC increase the users' privacy through the secrecy of the users' identities, better management of their attributes, and the authenticity and integrity of the information's flows. The notarial authority contributes to increasing the security of the ARAs by limiting the ARAs' communication, which only the NAs would be able to request to verify the users' attributes. The NA also provides the users' attributes unlinkability, i.e., the SP can not link the user's identity inside the AAA with his attributes each time or each different services he accesses with different AAA (if the user determines a different *id* for each AAA). The ARA can not trace the user by analyzing each time the SP requests the user's attributes verification. The TSL manages the trust of the existed NAs and ARAs, keeping up to date their information and their public keys.

Anonymity and other privacy characteristics are also satisfied by the notarial authority, which is a trust entity and their policies must keep the security of the user's information during the procedures. The anonymity authentication procedure, through a *nonce* created and ciphered by the NA, provides the authenticity of the AAA sent by the user and the acknowledgement of the SP to confirm that the AAA was created by the same user with whom it is communicating. The AAA's signature done by the user (at the moment when the AAA is created) provides the authenticity, the integrity, and the non-repudiation, about the user's attributes claimed by himself. The signature made by the NA, co-signing the AAA, results in the veracity confirmation of the information claimed by the user, and that the attributes are binded to the cyphered user's master public key.

The user might store some AAAs already co-signed by the NA to speed up the process of requesting a resource to SP. With a co-signed AAA, the user could access a resource in an off-line mode, i.e., physically in the real world. To facilitate the AAAs' management, we assume that an application should be used to store the co-signed AAAs in a mobile device (with a secure mechanism) and the users' master secret key stored in a token and plugged into the device only when requested.

As a consequence of the ciphered *nonce* that is exchanged between the user and a service provider, each co-signed AAA works for a specific SP due to the *nonce* ciphered with the SP's public key. Another consequence of the proposed model is the transition of the responsibility's control of the attribute disclosed to their owners. It is important that the users being aware of how they should protect themselves when communicating with a service provider.

Differently from the traditional, already known, identity and access management systems, e.g., OpenID and SAML-based (like the Shibboleth framework [42]), the principal technology used in our model is the asymmetric cryptographic functions and it could also work in a non-web environment. Additionally, we do not propose a specific standard to be used in the communication's workflow neither we specify which technology must be used to implement the system. We only determine the paradigm, the concepts, the necessaries cryptographic functions, and letting the developer to decide which technology best fit for his implementation.

The differences between the UCPKI, Idemix and U-Prove user-centric approaches, mainly differ at the architecture. In the Idemix and U-Prove architectures, each attribute provider should be a credential issuer and there will be necessary a user authentication mechanism (e.g., login and password) to request the credential. The UCPKI one is based on notary, which it is responsible to communicate with the correspondent attribute provider to validate the user's attributes. Idemix and U-Prove are selective disclosure approaches, which many user's attributes are included into a smartcard and then, the user decides which ones will be disclosed at each use. At the UCPKI approach, each assertion has only those attributes that are going to be disclosed to that specific service provider. This approach provides a freshness of the user's attributes because the assertion does not need to have a long term validity.

VII. Considerations and Future Work

The use of the standard X.509 PKCs allows multiple digital processes becoming more secure for entities and information involved. However, this mechanism does not take into account the management of the users' attributes and their privacy. We presented a model that increases the way that users control and disclose their personal attributes. The UCPKI architecture aims to eliminates the complexity and problems caused by the PKI and PMI standards. The users' privacy is enhanced by the use of identity-based cryptography and the user-centric paradigm.

Based on the notaries' responsibilities, the notarial authorities validate the users' attributes communicating with the responsible attribute registration authority. The NAs increase the workflow and the users' privacy. Differently from other identity and access management infrastructures, UCPKI keeps the strength of the cryptography's functions and the dynamism of the IBC to simplify the authentication and authorization infrastructure. Additionally, UCPKI is less costly to end-users compared to PKI. For future works, we suggest a calculation of the processing necessities and the capabilities to focus in ubiquitous computing and environments. Moreover, the UCPKI model could be also applied in documents signatures procedures, and a description of the notarial authority validation procedures of the user's attributes and signature is needed to be compared with the PKCs ones.

 ISBN: 978-1-61208-305-6

REFERENCES

[1] D. Cooper *et al.*, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 5280, Internet Engineering Task Force, May 2008.

[2] W. Diffie and M. Hellman, "New directions in cryptography," *Information Theory, IEEE Transactions on*, vol. 22, no. 6, pp. 644–654, Nov. 1976.

[3] P. Gutmann, "PKI: It's Not Dead, Just Resting," *Computer*, vol. 35, no. 8, pp. 41–49, Aug. 2002.

[4] A. Lioy, M. Marian, N. Moltchanova, and M. Pala, "PKI Past, Present and Future," *International Journal of Information Security*, vol. 5, pp. 18–29, 2006.

[5] C. Adams and M. Just, "PKI: Ten Years Later," in *In 3rd Annual PKI R&D Workshop*, 2004, pp. 69–84.

[6] D. Berbecaru, A. Lioy, and M. Marian, "On the Complexity of Public-Key Certificate Validation," in *Information Security*, ser. Lecture Notes in Computer Science. Springer Berlin / Heidelberg, 2001, vol. 2200, pp. 183–203.

[7] S. Farrell, R. Housley, and S. Turner, "An Internet Attribute Certificate Profile for Authorization," RFC 5755, Internet Engineering Task Force, Jan. 2010.

[8] ITU-T, "Information Technology – Open Systems Interconnection – The Directory: Public-key and Attribute Certificate Frameworks," International Telecommunication Union - ITU, Tech. Rep., Nov. 2008, International Standard ISO/IEC 9594-8.

[9] T. P. Hormann, K. Wrona, and S. Holtmanns, "Evaluation of Certificate Validation Mechanisms," *Computer Communications*, vol. 29, no. 3, pp. 291–305, 2006.

[10] K. Scheibelhofer, "PKI without Revocation Checking," in *4th Annual PKI R&D Workshop*, NIST, Ed., 2005, pp. 48–61.

[11] M. Ofigsbo, S. Mjolsnes, P. Heegaard, and L. Nilsen, "Reducing the Cost of Certificate Revocation: A Case Study," in *Public Key Infrastructures, Services and Applications*. Springer Berlin Heidelberg, 2010, vol. 6391, pp. 51–66.

[12] C. T. Moecke, R. F. Custódio, J. G. Kohler, and M. C. Carlos, "Uma ICP Baseada em Certificados Digitais Autoassinados," in *SBSeg*, Fortaleza-CE, Brazil, 2010, pp. 91–104.

[13] M. A. G. Vigil, R. F. Custódio, N. da Silva, and R. Moraes, "Infraestrutura de Chaves Públicas Otimizada: Uma ICP de Suporte a Assinaturas Eficientes para Documentos Eletrônicos," in *SBSeg*, Campinas-SP, Brazil, 2009, pp. 129–142.

[14] M. A. G. Vigil, C. T. Moecke, R. F. Custódio, and M. Volkamer, "The Notary Based PKI – A Lightweight PKI for Long-term Signatures on Documents," in *EuroPKI*, Sep. 2012.

[15] C. Adams and R. Zuccherato, "Notary protocols," Internet Draft, Tech. Rep., 1997. [retrieved: Oct., 2013]. Available: http://tools.ietf.org/html/draft-adams-notary-01

[16] Z. Chao-yang, "An improved computer notary system protocols," in *Intelligence Information Processing and Trusted Computing (IPTC), 2011 2nd International Symposium on*, 2011, pp. 242–244.

[17] C. Ellison *et al.*, "SPKI Certificate Theory," RFC 2693, Internet Engineering Task Force, Sep. 1999.

[18] T. Saito, K. Umesawa, and H. Okuno, "Privacy enhanced access control by SPKI," in *Parallel and Distributed Systems: Workshops, Seventh International Conference on*, Oct. 2000, pp. 301–306.

[19] R. L. Rivest and B. Lampson, "SDSI – A Simple Distributed Security Infrastructure," Apr. 1996.

[20] A. Jøsang and S. Pope, "User Centric Identity Management," in *In Australian Computer Emergency Response Team Conference*, 2005.

[21] OASIS, "Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS Standard, Oct. 2005. [retrieved: Oct., 2013]. Available: http://docs.oasis-open.org/security/saml/v2.0/saml-conformance-2.0-os.pdf

[22] OpenID, "OpenID Authentication 2.0 - Final," Dec. 2007. [retrieved: Oct., 2013]. Available: http://openid.net/specs/openid-authentication-2_0.html

[23] OASIS, "Web Services Federation Language (WS-Federation) Version 1.2," OASIS Standard, 2009. [retrieved: Oct., 2013]. Available: http://docs.oasis-open.org/wsfed/federation/v1.2/os/ws-federation-1.2-spec-os.pdf

[24] H. Nogueira, D. B. Santos, and R. F. Custódio, "Um Survey sobre Ferramentas para Single Sign-On," in *Workshop de Gestão de Identidades - WGID/SBSeg*. Brazil: WGID/SBSeg, 2012, pp. 522–542.

[25] OASIS, "Oasis WS-Trust 1.4," OASIS Standard, Apr. 2012. [retrieved: Oct., 2013]. Available: http://docs.oasis-open.org/ws-sx/ws-trust/v1.4/ws-trust.pdf

[26] A. Pfitzmann and M. Hansen, "A terminology for talking about privacy by data minimization: Anonymity,Unlinkability, Undetectability, Unobservability, Pseudonymity, and Identity Management," Aug. 2010, v0.34. [retrieved: Oct., 2013]. Available: http://dud.inf.tu-dresden.de/literatur/Anon_Terminology_v0.34.pdf

[27] R. Leenes, J. Schallaböck, and M. Hansen, "Prime white paper," *PRIME (Privacy and Identity Management for Europe), White Paper*, 2008. [retrieved: Oct., 2013]. Available: https://www.prime-project.eu/prime_products/whitepaper/PRIME-Whitepaper-V3.pdf

[28] J. Angulo, S. Fischer-Hübner, E. Wästlund, and T. Pulls, "Towards usable privacy policy display & management for primelife," *Inf. Manag. Comput. Security*, vol. 20, pp. 4–17, 2012.

[29] J. Camenisch and A. Lysyanskaya, "An Efficient System for Non-transferable Anonymous Credentials with Optional Anonymity Revocation," in *Proceedings of the International Conference on the Theory and Application of Cryptographic Techniques: Advances in Cryptology*, ser. EUROCRYPT. Springer-Verlag, 2001, pp. 93–118.

[30] C. Paquin, "U-prove technology overview v1.1," Microsoft Corporation, Tech. Rep., April 2013. [retrieved: Oct., 2013]. Available: http://research.microsoft.com/pubs/166980/U-Prove%20Technology%20Overview%20V1.1%20Revision%202.pdf

[31] M. Joye and G. Neven, *Identity-Based Cryptography*. Ios PressInc, 2009, vol. 2.

[32] J. Oltsik, "The True Costs of E-mail Encryption," Enterprise Strategy Group, White Paper, 2010. [retrieved: Oct., 2013]. Available: http://www.trendmicro.de/media/ds/email-encryption-costs-esg-whitepaper-en.pdf

[33] A. Kumar and H. Lee, "Performance Comparison of Identity Based Encryption and Identity Based Signature," *International Journal of Security and Its Applications*, vol. 6, no. 3, pp. 19–28, 2012.

[34] D. Boneh and M. Franklin, "Identity-Based Encryption from the Weil Pairing," in *Advances in Cryptology – CRYPTO 2001*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2001, vol. 2139, pp. 213–229.

[35] C. Cocks, "An Identity Based Encryption Scheme Based on Quadratic Residues," in *Proceedings of the 8th IMA International Conference on Cryptography and Coding*. Springer-Verlag, 2001, pp. 360–363.

[36] J. Baek, J. Newmarch, R. Safavi-naini, and W. Susilo, "A Survey of Identity-Based Cryptography," in *Proc. of Australian Unix Users Group Annual Conference*, 2004, pp. 95–102.

[37] A. Sahai and B. Waters, "Fuzzy Identity-Based Encryption," in *Advances in Cryptology – EUROCRYPT 2005*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2005, vol. 3494, pp. 457–473.

[38] V. Goyal, O. Pandey, A. Sahai, and B. Waters, "Attribute-based Encryption for Fine-grained Access Control of Encrypted Data," in *Proceedings of the 13th ACM conference on Computer and communications security*, ser. CCS '06. ACM, 2006, pp. 89–98.

[39] S. Shahandashti and R. Safavi-Naini, "Threshold Attribute-Based Signatures and Their Application to Anonymous Credential Systems," in *Progress in Cryptology – AFRICACRYPT 2009*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2009, vol. 5580, pp. 198–216.

[40] ETSI, "Electronic Signatures and Infrastructures (ESI); Provision of Harmonized Trust-service Status Information," Tech. Rep. TS 102 231, Dec. 2009.

[41] N. Haller, C. Metz, P. Nesser, and M. Straw, "A One-Time Password System," RFC 2289, Internet Engineering Task Force, Feb. 1998.

[42] H. Nogueira, R. F. Custódio, C. T. Moecke, and M. S. Wangham, "Using Notary Based Public Key Infrastructure in Shibboleth Federation," in *Workshop de Gestão de Identidades - WGID/SBSeg*. Brazil: SBSeg, 2011, pp. 405–414.

 ISBN: 978-1-61208-305-6