Implementing a Privacy-enhanced ABC System for Online Social Networks with Co-Ownership Management

Esther Palomar[1]*, Lorena González-Manzano[2], Almudena Alcaide[3] and Álvaro Galán[2]

[1] Birmingham City University, UK. Email: esther.palomar@bcu.ac.uk

[2] University Carlos III of Madrid, Spain. Email: lgmanzan@inf.uc3m.es

[3] IT-ERS at Deloitte, Spain. Email: aalcaide@deloitte.es

May 12, 2015

Abstract

Online social network (OSN) users are exhibiting an increased privacy-protective behaviour specially since multimedia sharing has emerged as a popular activity over most OSN sites. Popular OSN applications could reveal much of the users' personal information or let it easily derived, so favouring different types of misbehaviour. In this article we deal with these privacy concerns by applying fine-grained access control and co-ownership management over the shared data. Our proposal defines access policy as any linear boolean formula that is collectively determined by all users being exposed in that data collection namely the co-owners. All co-owners are empowered to take part in the process of data sharing by expressing (secretly) their privacy preferences and, as a result, jointly agreeing on the access policy. Access policies are built upon the concept of secret sharing systems. A number of predicates such as gender, affiliation, or postal code can define a particular privacy setting. User attributes are then used as predicate values. In addition, by the deployment of Privacy-enhanced Attribute-Based Credential (Privacy-ABC) technologies, users satisfying the access policy will gain access without disclosing their real identities. We have implemented our system as a Facebook application demonstrating its viability, and procuring reasonable performance costs.

Keywords.- Online Social Networks, Privacy-aware Access Control, Anonymity, Co-Ownership Management, Attribute-based Credential System

1 Introduction

Nowadays, privacy is globally considered a very valuable personal asset. However, we find ourselves, more and more often, spreading most of our personal details, eating habits, work life etc. across many different domains and online scenarios [1]. Indeed, online social networks (OSNs) get hold of a huge amount of personal data which, in the majority of cases, even if uploaded by a particular user, involves several ones. In this context, challenging privacy issues arise from co-ownership of shared personal data, since tagging is a common technique. Users tag a particular piece of data, e.g. a photo or any other resource, with a set of individual names leaving access control policies out of the co-owners control. As a result, the owner (here considered as the uploader as well) does choose between the available set of different granularities most in the way of options like Only Me, Friends Only, etc. or by providing user-defined friend lists. These common controls range from limitation of profile access, item-level access control, to other features such as blocking and hiding other site users, however, do not consider or evaluate other co-owners' preferences that could be conflicting between each other.

In this work, we aim to demonstrate the suitability of quite robust cryptography primitives to address some of the foregoing privacy issues in the context of OSNs. In particular, these are the primary contributions of this article:

- *Co-ownership management of shared data collections.* Our aim is at empowering not only the data owner but also the co-owners to take part in the process of expressing (secretly) their privacy preferences of the shared data to be published and, as a result, jointly agreeing on its access policy. Access policy is then collectively determined by all users exposed in that data collection while, at the same time, individual privacy preferences are kept hidden. To encounter this challenge, a method for the collectively establishment of the privacy settings for shared data in OSNs can be built upon the concept of threshold secret sharing [2].

- *Fine-grained access control management.* We represent access policies as any linear boolean formula (or a conjunction of formulas) which compile every co-owner's privacy preferences over the shared data. A number of predicates such as gender, age, affiliation, nationality or postal code can define a particular data privacy setting. User attributes are then used as predicate values for fine-grained access control management.

- *Anonymous access to shared data.* OSN users who prove that they satisfy the access policy will gain access to those shared resources without disclosing their real identities. Therefore, users hold a certified list of attribute-values which in the way of a credential will be used to make statements about themselves (or a subset of their attributes) anonymously and in a certified manner. The application of a privacy-enhanced Attribute-Based Credential (ABC) system will assure anonymous and unlinkable authentication and data access control [3].

Our proposal then combines threshold secret sharing and anonymous ABC algorithms into a novel model for a privacy-aware management of the shared data access and co-ownership in OSNs. In particular, Joint Random-Secret Sharing (JRSS) [2] will provide co-owners executing the co-ownership management with the confidentiality of their privacy preferences. The access to OSNs data will be also prevented from profiling, gossiping and other privacy concerns with social networking by the inclusion of the ABC System presented in [3] which is based on U-Prove [4] for authorization purposes. We develop a Facebook application called CANONYM (Co-ownership and Anonymity) to evaluate the feasibility of our proposal.

The reminder of this article is organized as follows. Section 2 discusses the related work. Section 3 provides a brief overview of our system components and building blocks. We further elaborate on each system phase in Section 4. System implementation is described in 5. Finally, in Section 6 we establish the main conclusions.

2 Related Work

Related work has been investigated in three different areas: co-ownership or collaborative privacy management, access control and anonymity in OSNs.

2.1 Co-ownership Management in OSNs

Communication Privacy Management theory states that the decision to disclose private information to others involves negotiating expectations about how the information will be collectively managed once divulged [5]. However, collective privacy management in OSNs has been sparse so far and headed by very simple voting-based approaches [6, 7].

A. Squicciarini and H. Hu are the initiators of co-ownership management. Squicciarini *et al.* present a Clarke-Tax algorithm to take collective privacy decisions about the access control policy to apply which are based on path distances. Later on, Squicciarini *et al.* introduce CoPE, an application to promote collaborative specification of policies focused on the most highly voted option [6]. Similarly, a multiparty authorisation framework for Facebook-style social networks is developed by H. Hu *et al.* in [7] and significantly extended in [8]. Authors propose a multiparty policy specification scheme defining the logical representation of access control policies, as well as a policy evaluation mechanism which deals with policy conflicts by keeping the balance between the need for privacy protection and the users' desire for information sharing. Evaluation of a prototype implementation showed a measurement of average expected privacy risk and sharing loss.

B. Carminati and E. Ferrari introduce the term *collaborative security policies* [9]. Data owners establish access control policies taking into account distance and trust of their contacts' relationships. These policies are sent to co-owners asking for feedback.

Previous approaches compromise some users' privacy, however, there are a couple of exceptions. Work in [10] allows the establishment of access control policy when all users reach a full consensus. Moreover, [11] presents a mechanism based on decomposable image-based objects to guarantee, in a fine-grained way, all users' privacy.

2.2 Cryptographic Approaches to Privacy and Access Control in OSNs

Significant work has been done exploring cryptographic approaches to enhance the content sharing privacy on OSNs. Work in [12] presents a public-key protocol which achieves relationship protection without the presence of a central node so enabling private relationships using certificates or verifiable credentials.

A number of the proposed access control models leverage users' attributes. These attributes, e.g. relationships, roles, or other contextual information, can be used to aid users in configuring their settings and expressing their privacy preferences with a fine-granularity [13]. There are some other works also ensuring access control in OSNs through public key cryptography and attribute-based encryption (ABE) over group members, mainly considering a fully distributed approach [14, 15]. In [15] a novel cryptographic primitive based on ABE and relationship links is introduced. Authors introduced an access control framework and implemented it as a

Facebook application without having to trust the OSN manager. Similar to Persona framework [14], attribute conditions are applied to specify access policies concerning the protected data. However, the set of attribute values are not prevented from being inferred making user transactions traceable, linkable and observable.

2.3 Approaches to Anonymity in OSNs

User anonymity in OSN has been addressed mainly by preventing topology-based attacks and focusing on the recognition of the social network structure [16]. For instance, A. Campan *et al.* proposed the generalization of a pair of clusters of OSN users where users become indistinguishable [16].

Furthermore, system called Gossple in [17] associates every OSN user with a set of anonymous acquaintances who share common interests, whereas Pisces presented in [18] anonymizes communications, most in the way of TOR.

Our approach has common points with some recent approaches such as [19] which apply authorization modalities that do not sacrifice user anonymity by the use of zero-knowledge proofs of knowledge (ZKPoK). Pedersen Commitment Scheme is employed by N. Shang *et al.* in [19] to develop a novel attribute-based access control mechanism for protecting content dissemination (in health care applications). Besides, with regard to decentralization, other works have applied threshold-based secret sharing protocols for addressing different problems in OSNs [20].

2.4 Discussion

Table 1 depicts a summary of the analysis to the related work. First, there is a general lack of fine-grained access control management and definition of privacy policies mostly due to existing work bases on direct or indirect relationships, or, at most, n-hop distanced. Both co-ownership and fine-grained access control are addressed in [11] where objects are composed of parts. Owners assign each object's part to the appropriate user who becomes a co-owner and who specifies attribute-based access control policies. Access to each part is independently granted or denied according to policies established by the associated/ assigned user (the owner or a co-owner). Likewise in H. Hu *et al.*'s work [8], voting strategies have also been applied to aggregate an obtained score and, by establishing thresholds, access becomes denied, granted, or restricted to a set of users (e.g. *owner-overrides, full-consensus-permit, majority permit*).

Table 1: Analysis of the related work.

	Types of policies	**Secret specification of preferences**	**Manage co-ownership**	**Anonymity of**
A. Campan *et al.* [16]				OSN users
M. Prateek *et al.* [18]				Communications
M. Bertier *et al.* [17]				OSN users
A. Squicciarini *et al.* [21]	owners/ n-distance users/ public	√	Clarke-Tax algorithm	
A. Squicciarini *et al.* [6]	co-owners only/ some-friends/ public	√	The most voted option	
H. Hu *et al.* [7] [8]	owner-overrides/full-consensus-permit/ majority permit/ strong-majority permit/ super-majority permit		Score based on sensitivity level and its aggregation	
B. Carminati *et al.* [9]	n-distance users/ trust		One, all or the most voted option	
K. Thomas *et al.* [10]			Full consensus	
González-Mazano *et al.* [11]	Based on attributes conditions and obligations		Each user manages their objects' parts	
R. Baden *et al.* [14]	Based on attributes			
S. Jahid *et al.* [22]	Based on attributes			
S. Braghin *et al.* [23]	Based on attributes			

Secondly, Squicciarini *et al.* proposals point out the secretly specification of preferences. In particular, work in [6] proposes a voting scheme in which by default, co-owners (called stakeholders) have no knowledge of other users preferences. Co-owners specify their preferences (namely some-friends, public or co-owner) and the most voted option is the established policy. However, there is no further elaboration on the way of that secrecy is performed, nor explanation if, at some point, an entity realised about other users' preferences. Thus, anonymity and co-ownership are independently managed without considering the significance of their joint application [24].

3 Background and System Overview

We describe all necessary components, roles and phases of the proposed system.

3.1 Joint Random-Secret Sharing

JRSS has been extensively studied and used in threshold cryptography and secure multiparty computation (see details [2, Section 6.3]). Besides, the implications of deploying threshold-based secret sharing in a distributed OSN have been recently tackled in [25]. The essential notion to this scheme is that a number of participants comes together to generate a random-secret piece of information in a unanimous consensus manner by contributing each's private input. Without the assistance of any third trusted party, each participant's input has an equal influence on the determination of the secret whereas this is kept undisclosed.

3.2 Anonymous Attribute-Based Credentials

ABC systems provide a friendly privacy-preserving mechanism to minimize the amount of personal data disclosed during authentication and authorization processes whilst ensuring correctness of the data.

Various interesting aspects of ABC systems make them very suitable to be used in practice in current OSNs:

- An ABC is a certified list of attribute-values. The main idea behind anonymous ABC is that users can prove statements about themselves (a subset of their attributes) anonymously and in a certified manner.
- A number of predicates (boolean expressions built applying logic, comparison and arithmetic operations to attribute values) over the ABC attributes can be formulated and then used to evaluate if certain conditions are satisfied for a particular access control policy, e.g. only teenagers are authorized according to the ID card/passport.
- Anonymous ABC systems are based on the concept of ZKPoKs [26]. Users of anonymous ABCs are able to prove, to a verifying entity, holdership of the credential, knowledge of all attribute values or that such values satisfy a given property (such as belonging to a range or satisfying a function) without revealing the attribute values themselves.

Our anonymous ABC system for OSNs is based on Persiano et al. Anonymous Credential System [3] which is based on U-Prove [4]. In this system, users must prove on a zero-knowledge fashion that they know the value of their hidden attribute values but also equally important is that those values satisfy a linear relation specified in a boolean formula. Furthermore, as in the Idemix [27], Persiano et al. system supports the following properties: (i) *Issuance-show unlinkability*[1] by which the ABC issuer cannot recognize credentials when they take part in showing protocols and, (ii) *Multi-show unlinkability* by which multiple showings of the same ABC cannot be linked.

[1] The ABC system in [3] preserves issuance-show unlinkability, this is, the authority issuing the credential cannot link the credential issued with the credential being shown to the verifying entity. The system also offers multi-show unlinkability so different uses or shows of the same credential can be linked together).

In this article, the process that a user must follow to obtain and use an ABC is described in Sections 4.2 and 4.3 respectively, although leaving the description of the mathematical foundations and formal proofs to the cited publications.

3.3 Roles and Phases

The system defines two non-mutually exclusive roles for OSN users and is divided in three phases, namely

- *Co-Ownership Management.* The *Originator* is an OSN user who wants to control the access to some shared resource which is co-owned by one or more *Co-owners.* In general, the Originator and Co-owners know each other and are connected via the OSN site. They all will gather together for establishing the access control policy for that resource. No one in the process will be able to realise about others' privacy preferences for the resource.

- *ABC Acquisition.* A central organization called *Issuer* is in charge of providing users with their corresponding anonymous ABCs. The *User* role is then granted during this phase by asking the Issuer for a valid ABC.

- *Resource Access.* Users, who want to access such shared resource, need to anonymously and unlinkably prove possession of a valid ABC encoding a particular set of attributes satisfying the access control policy. The resource is published by a *Provider*; this is either the Originator, or any other User or Co-owner who had got previously access to the resource and published it afterwards.

4 Proposed System Phases

We illustrate at a high-level our privacy-enhanced ABC system phases, and some technical details are omitted (we refer the reader to the corresponding literature for further mathematical foundations).

4.1 Co-Ownership Management Phase

In this phase, co-owners of a shared resource gather together for the generation of its privacy policy at the request of the Originator. The privacy policy of a common shared data is represented as any linear boolean formula (or a conjunction of formulas) which compiles every co-owner's privacy preferences over that data. Hence, a number of predicates such as status, gender, age, departmental affiliation, nationality and postal code, to name a few, will define a particular data privacy setting.

Example: Oscar uses an OSN to keep in touch with people he knows and decides to share a particular collection of photos about his new physical skills of fitness at the gym. He is only interested in sharing these photos with his friends from the neighborhood being currently over 18. Obviously, the predicate "age$>$ 18" can be activated to filter minors and, so on so forth. However, four more people appear in the photos who are members of the same health club that Oscar frequents and, have got user accounts[2] in the same OSN site. Thus, instead of uploading the photo collection with his own privacy policy, Oscar enrolls these four friends into the execution of the *Co-Ownership Management* phase.

Without loss of generality, our proposal is described assuming that users are authenticated based on the values of their identity attributes $(x_1, \cdots, x_k)$, which can be derived from a standard certificate signed by some trusted external entity (i.e., a certification authority). Therefore we represent a privacy policy of a shared resource ξ as the boolean formula $\Phi_\xi(x_1, \cdots, x_k) = 1$ whose inputs are the authentication attributes of a user. It combines a series of boolean formulas specifying the privacy preferences whose relations are linear in those attributes, i.e., such formulas are connected by $\wedge, \vee, \neg$ logical connectives.

For the sake of illustration, considering the predicate "age$>$ 18" defined above there exists a linear equation on two attributes (x_1, x_2) which can be derived from a valid passport such that $x_1 = 6$ and $x_2 = 24$ of a user aged 24 can be proven to satisfy the equation $x_1 + 18 = x_2$, i.e. the user is over 18. Our proposal can be easily extended to any number of attributes and the semantics[3] may differ from the one in the example. For instance, from a propositional logic point of view, an access control rule can be represented by a linear boolean formula over the specific predicates (or a function on them) and their relations, for a certain resource. The semantics attached to defining predicates should be adapted to each specific case. As a particular

[2] We assume that other participants have to be registered in the OSNs to be part of the system.

[3] We also consider that, either by a third party or even crowdsourcing, the attribute 'Friend of' can be certified and then used as user attribute as well as to build policy predicates.

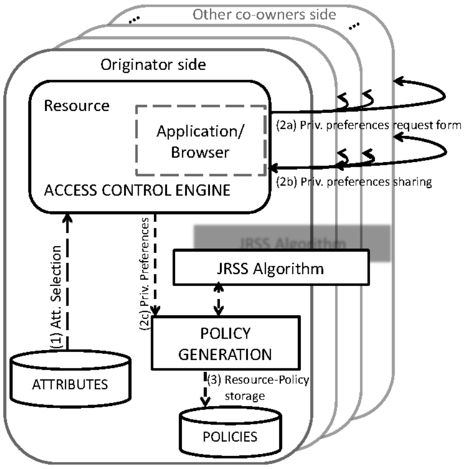


Figure 1: Joint establishment of the privacy preferences.

setting, the following formula:

$$(x_1 + 18 = x_2) \wedge (\neg x_3 = \text{“BCU”}) \wedge \big((x_4 + 28000 = x_5) \vee (x_4 + 8000 = x_5)\big)$$

defines a privacy policy in which the co-owners agree on grating access to over-18 users, not affiliated with "BCU" and registered in Madrid (postcode range 28000-28999) or Barcelona (postcode range 8000-8999).

The reasons behind the generation of these parameters in a joint and secret manner are mainly two: (1) Resource co-owners can secretly impose privacy preferences on the access to their private common resources (co-owners will learn nothing about each others' preferences) and, (2) the resulting parameters are the combination of the preferences of all co-owners which maximizes the global social utility and promotes fairness encouraging truthfulness among the co-owners.

Figure 1 is a conceptual depiction of our *Co-Ownership Management* phase. Thus, considering the particular scenario to filter adults such that "age> 18" (step 1), the originator starts the process by asking the other $n-1$ co-owners of the collection of data to be published, for secretly sharing their privacy preferences over that predicate (step 2a). In practice, this predicate can be represented as a binary variable being encoded numerically either by 1 so activating the filter or 0. Each co-owner contributes its preference (which is kept confidential thanks to the JRSS algorithm) over that predicate and sends it over to the Originator/server side (step 2b). Once all co-owners' preferences are compiled, the policy is generated, and stored in the server along with the protected collection of data depending on the JRSS output (steps 2c and 3, respectively). More precisely, a restrictive policy will be applied to the shared data if the output jointly computed falls within the range $[\frac{n}{2}, n]$. Otherwise, values within the interval $[0, \frac{n}{2})$ leave the data accessible, with no age control. We refer the interested reader to [2] and to our implementation described in Section 5. For the sake of completeness, the basics steps for the JRSS to share a secret integer over $\mathbb{Z}_q$ (for any q prime) are detailed in the pseudocode below.

Algorithm 1 JRSS algorithm over $\mathbb{Z}_q$ (for any q prime)

```
Require: Co-owner $i$ generates a $t$-degree polynomial $f$
Ensure: $f(0) = \mathit{secret\ integer}$
  (Step 1) Originator picks attribute label
  (Step 2) Originator notifies Co-owners
  for all $i \in Co - owners - Set$ do
    $\{a_0^i, \ldots, a_t^i\} \in \mathbb{Z}_q \leftarrow t+1$ random values
    Construct the polynomial $f^i(x) = \sum_{k=0}^{t} a_k^i x^k$
    for all $j \in Co - owners - Set$ do
      Calculate value $f^i(j)$
      Unicast $f^i(j)$
      Collect $(t+1)$ values $\{f^j(i)\}$
    end for
    $f(i) \leftarrow \sum_j f^j(i)$
    (Step 3) Originator sets the policy $f(0) \leftarrow \sum_j a_0^j$
  end for
```

4.2 ABC Acquisition Phase

As previously mentioned, we have based our work on the ABC system by Persiano's et al. [3]. In their work, authors describe a series of five algorithms namely *SetUp*, *Enroll*, *IssueCred*, *ProveCred* and *VerifyCred* which conform the whole Credential System and which operate on a set of chosen global parameters (an algebraic group G, a subgroup of G of certain order and a series of generators). The mathematical operations performed (commitments and ZKPoKs) on those parameters and on the User's attributes are based upon the Problem of the Discrete Logarithm and, are designed to prove possession of an ABC, digitally signed by the corresponding authority, which encodes the correct attribute values.

We assume that such a credential system is set-up properly by a certain certification authority (namely the Issuer, which is not necessarily the same as the OSN service provider) which executes algorithm *SetUp* on input the publicly verifiable parameters[4] known as *Pub*. The Issuer also obtains the private information *Priv* which corresponds to the public information *Pub* that she will use to release user ABCs. Detailed information and mathematical properties of these two tuples are fully described in [3] and [28].

In particular, the proposed *ABC Acquisition* phase is based and inherits the instructions from the algorithms *Enroll* and *IssueCred* in [3, Section 4.2]. During this phase a User presents non privacy–aware credentials to the Issuer which is responsible for the verification and extraction of the attributes[5] $(x_1, x_2, \ldots, x_n)$ that will be encoded in attribute-based credential. The Issuer is also responsible for the construction and signature of the ABC. Hence, the ABC is not more than a digitally signed tuple of encoded values of the form:

$$ABC = \langle x_1, x_2, \ldots, x_n, x, z, v \rangle$$

where x_i references each of the User attributes, x and z are random numbers chosen by the Issuer and v is the signature of the Issuer over them. Thus, the Issuer constructs, from a series of non privacy–aware certificates, a new privacy–aware certificate.

In the example considered above, Oscar and his friends, after executing the *ABC Acquisition* phase, will come into possession of a signed ABC encoding the attributes that will be required and used to access the OSN site services and contents.

4.3 Resource Access Phase

Any User can be granted access to the shared resources iff his/her subset of attributes required satisfies the access policy.

The *Resource Access* phase is performed by a User (as the prover) requesting access to a particular resource ξ and a provider[6] (as the verifier) responsible for enforcing the access control policy of ξ. This phase is based

[4] Publicly Verifiable Secret Sharing plays an important role in the design of protocols for secure multi-party computation as everybody is able to verify that the shares have been correctly distributed.

[5] For the sake of completeness, note that $(x_1, x_2) \in \mathbb{Z}_e$ so $0 \leq x_1, x_2 < e$ being $e \in Pub$ such that e is prime, $e \neq 2$, $e \in \mathbb{Z}_n^*$, $gcd(e, 3) = 1$ and n is the modulus, product of two safe primes. We refer the interested reader to [3] for further details on the mathematical foundations of this credential system.

[6] The provider is a User who is currently publishing the resource and could or not be part of the Co-owners set.

and inherits the instructions from the algorithms *ProveCred* and *VerifyCred* in [3, Section 4.2]. However, a few enhancements to these phases have been developed and implemented to attain further features. For instance, during credential verification, the Provider gets response to challenges that prove that the attributes encoded in the User's ABC satisfy the boolean formula Φ_ξ which bases on linear relations amongst the attributes. The Provider cannot learn anything about attributes' actual values. That is, any friend connected to Oscar and/or his friends will have anonymous access to the shared contents if and only if her attributes being verified against the formula give true.

The overall phase is shown in Figure 2 which shows the high-level sequence of steps corresponding to phase *Resource Access*. Note that the User could hold one or more ABCs stored in a repository. Each ABC could be signed by a different authority and may encode different attributes. Similarly, the provider has control over a series of resources each one attached to an access control policy. The flow is as follows:

1. The User requests a particular resource ξ.

2. In steps (2a) and (2b), the Provider makes the corresponding access control policy known to the User $\Phi_\xi(x_1, x_2, \ldots, x_n)$. In the step (2c), the User then uses the *Policy-credential matcher* to find a suitable credential (if any) whose attributes fulfill such particular policy.

3. In step (3a), to avoid traceability and linkability, the User produces a randomized and unique version of the original ABC and constructs commitments on the values of such randomized credential and sends (step 3b) them to the verifier to be store (step 3c).

4. The User and the Provider engage in a number of four ZKPoK over the encoded attributes as witnesses. In this regard, the Provider sends messages (4a), (4b) as a series of challenges to the User. These challenges are randomly generated by the Provider.

5. In steps (5a) and (5b), the User generates the corresponding responses to those challenges and sends them to the Provider which is in charge of the verification process (step 5c).

The User must follow the steps above to accomplish such proofs as the key part of the anonymous ABC access control system[7]. As a result, the Provider is convinced of the following:

[7] By using such proofs, at the cost of one exponentiation per base it is possible to

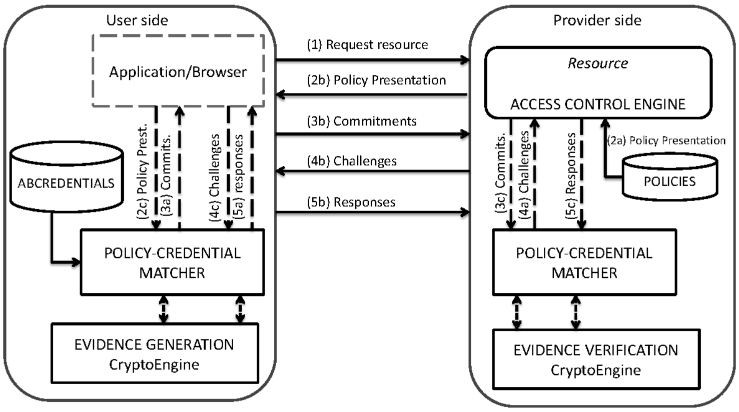


Figure 2: Resource Access phase. Our proposal bases on an ABC authentication and authorisation in the domain of OSNs where users are identified by a subset of their certified attributes that can be or not the same subset required for grating access to the data collections.

- The User holds an ABC such that their encoded attributes satisfy $\Phi_{\xi}(x_1, x_2, \ldots, x_n) = 1$. The verifier does not learn the values $(x_1, x_2, \ldots, x_n)$.
- The User has created fresh and correctly the series of commitments accompanying the ZKPoKs.
- The ABC held by the User includes a valid signature v, correctly constructed and signed by a valid Issuer.
- The randomized version of the ABC over which the computations are performed is fresh.

We refer the reader to [28, Section 3.3]-*Credential Proving* for a complete description of these ZKPoKs and to Section 5 where we present the implemented prototype details of this phase.

5 System Implementation

In order to evaluate the feasibility of the proposed system and of the related protocols, a Facebook (FB)-shaped application called CANONYM

prove in a witness indistinguishable manner knowledge of a discrete logarithm or of a DL-representation or of an RSA-representation [3].

(Co-ownership & Anonymity) has been developed[8] to provide FB users with co-ownership management and anonymous access to photo albums related to a particular topic and/or community. We consider a disable people community so shared content's privacy preferences can be bounded by being below or under 18, being or not a disabled person and being or not an European citizen. A simple web interface was created to simulate both Originator and Co-owner roles at the *Co-ownership Management* phase as well as Provider and User engaging in the *Resource Access*. The whole prototype hence was implemented as a mockup, with local hosting and databases.

Our evaluation was intended to cover basic aspects which may concern users aiming to carry out the functionalities described in previous sections:

- Time spent in the computation and communication by participants running the JRSS for the joint establishment of the access policy.
- Confidentiality of the privacy preference.
- Cost overhead (in terms of time) of the anonymous access to data namely time consumed in both computing the credential token and verifying the policy against it.

Furthermore, the following assumptions are noticed. The establishment of access control policies requires Originators to be logged in FB. By contrast, Users accessing the content may act out of FB. Moreover, co-owners have to be FB friends. Finally, Users are supposed to have already loaded at least an ABC on their Internet explorer, therefore the *ABC Acquisition* phase is not implemented for this piece of work.

5.1 System Architecture

As depicted in Figure 3, our development applies a client-server architecture and comprised the following modules:

- *FB server*: Its purpose is to store the resources along with resource names and other metadata.
- *CANONYM server*: Given restrictions to develop FB applications, this server comes into play. It consists of the following components:

[8] CANONYM prototype has been developed using J2EE and J2SE 1.6 over an Apache Tomcat v7.0.37 applying a MySQL data base.

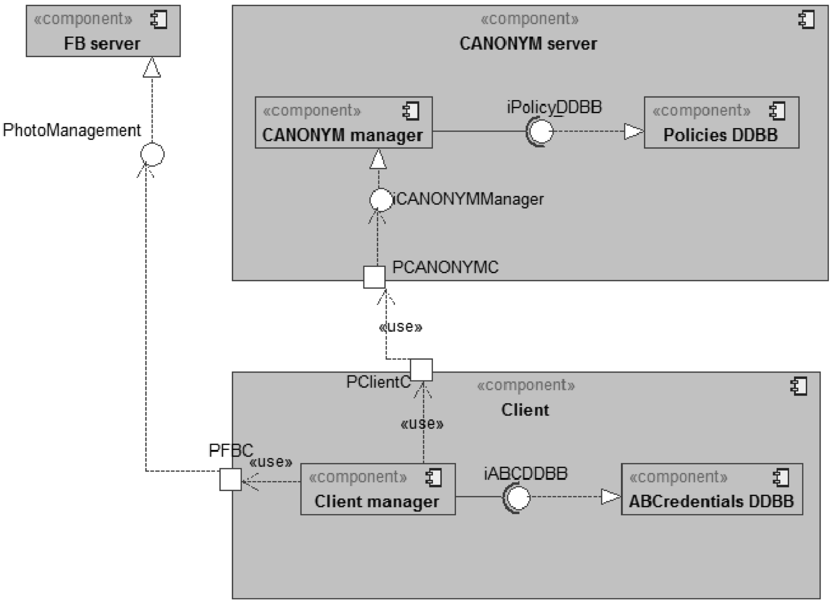


Figure 3: CANONYM architecture

The *CANONYM manager* executes two main tasks, namely the computation of the resource access control policies and the verification of policies upon incoming access requests.

Concerning the former task, this component computes a joint policy for a given photo album based on the Originator and $(t+1)$ Co-owners privacy preferences. A list of attributes and/or predicates are available in the server and can be easily selected by the Originator to establish the policy predicates considered in the policy generation. Moreover, assuming that all parameters are computed locally to each Co-owner, *CANONYM manager* cannot infer individual user preferences (see Section 4.1).

Regarding the policy verification, *CANONYM manager* is in charge of verifying a requester's ABC satisfies the policy associated with the requested resource. If the process in Section 4.3 succeeds, the access is granted and the appropriate link to the FB content is delivered, otherwise the access is denied.

Policies DDBB is a set of databases for the storage for the partial computations meanwhile the JRSS is being executing and final retrieval of resources' access control policies.

- *Client*: A couple of components are identified within this module:

  Client manager This module is mainly devoted to the computation of Users' contribution to the establishment of the shared data access policy, and the computation of the responses to the challenges sent by the Server when verifying policies.

  Concerning the computation of a data policy, each User specifies its personal privacy preferences that are sent to *CANONYM server* in such a way that the disclosure of individual users preferences remains infeasible. With regard to the access requests, the *Client manager*, using stored ABCs provides the necessary data to prove policies verification.

  ABCredentials DDBB is a database to manage the storage and retrieval of ABCs from the local repository.

- The system also needs an anonymizer (e.g. TOR[9]) to prevent attacks occurred at the network level.

With a web-based interface, our prototype allows us to experiment with the functions implemented at both the *Co-ownership Management* and *Resource Access* phases. The interface was kept simple as to access easily to the databases of users, attributes/predicates, policies, ABC and partial computations.

5.2 Co-ownership Management Implementation

The implementation of the *Co-ownership Management* phase is shown in pseudocode. This phase is initiated by the Originator at the manager side who chooses an album, selects the Co-owners involved and starts JRSS execution as follows:

1. The Originator selects preferences (x_i) namely age, nationality and disability, and, as a result, $(t-1)$ random values a^i are created per managed attribute, where t refers to the number of users involved in the process (Alg. 2 lines 1-10). Note that values a^i are created in the manager and sent to the client.

[9] https://www.torproject.org/, last access June 2013

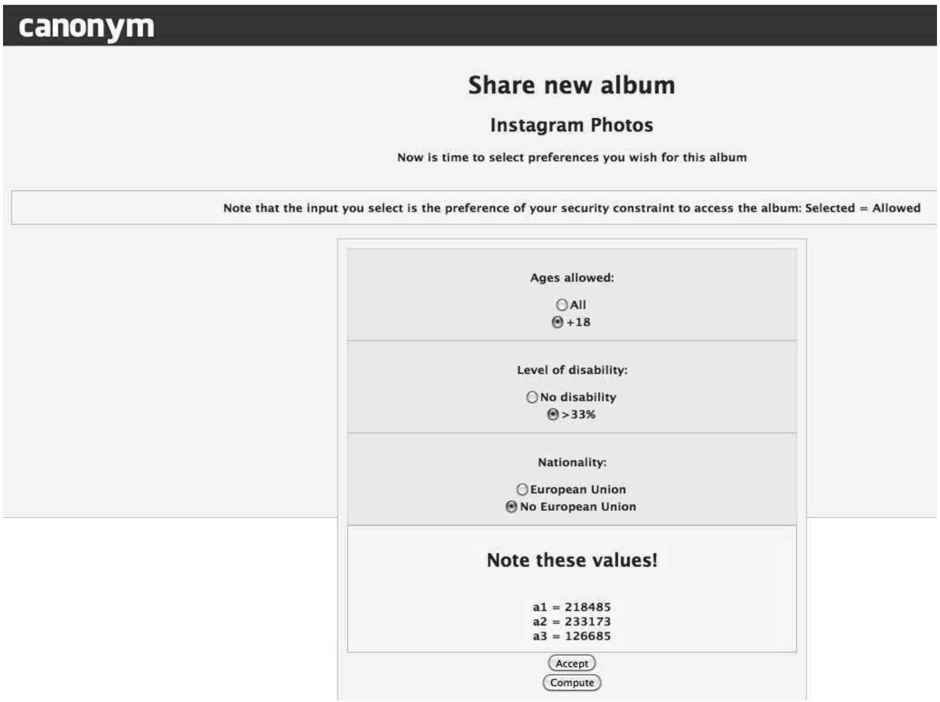
Algorithm 2 Polynomial of each user

```
 1: array a
 2: for i=0:AllUSers do
 3:     aAtt1[i]=random() {Att1 refers to AGE}
 4: end for
 5: for i=0:AllUSers do
 6:     aAtt2[i]=random(){Att1 refers to NATIONALITY}
 7: end for
 8: for i=0:AllUSers do
 9:     aAtt3[i]=random(){Att1 refers to DISABILITY}
10: end for
11: array valuePoly
12: for i=0:AllUsers do
13:     for j=0:aAtt1.length() do
14:         valuePolyAtt1[i]=aAtt1[j] * Pow(idsUsers[i],j+1) {Pow(x,y) means x is raised by the y
            power}
15:     end for
16: end for
17: for i=0:AllUsers do
18:     for j=0:aAtt2.length() do
19:         valuePolyAtt2[i]=aAtt2[j] * Pow(idsUsers[i],j+1)
20:     end for
21: end for
22: for i=0:AllUsers do
23:     for j=0:aAtt3.length() do
24:         valuePolyAtt3[i]=aAtt3[j] * Pow(idsUsers[i],j+1)
25:     end for
26: end for
27: UserId=n
28: for i=0:idsUsers.length() do
29:     if n=idUsers[i] then
30:         array value
31:         Store valuePolyAtt1[i]
32:         Store valuePolyAtt2[i]
33:         Store valuePolyAtt3[i]
34:     end if
35: end for
```

2. For each attribute and Co-owner, $f'(x) = \sum_{k=0}^{t}(a_k^i \cdot x^k)$ is constructed accordingly (Alg. 2 lines 11-26). Figure 4 depicts CANONYM interface for the establishment of privacy preferences.

3. The value of the polynomial in the Originator's ID is locally stored in the form of a vector and $f^j(j)$ for each Co-owner's ID (j) are sent to the manager to be stored in the DDBB (Alg. 2 lines 27-35).

4. A FB private message is sent over to each Co-owner requesting specification of their personal preferences, so repeating steps 1-3. Once the last Co-owner defines his preferences, the rest of them and the Originator are notified to compute the final access control policy, that is, the final polynomial. To do this, each User introduces the stored



Share new album

Instagram Photos

Now is time to select preferences you wish for this album

Note that the input you select is the preference of your security constraint to access the album: Selected = Allowed

Ages allowed:
○ All
● +18

Level of disability:
○ No disability
● >33%

Nationality:
○ European Union
● No European Union

Note these values!

a1 = 218485
a2 = 233173
a3 = 126685

Figure 4: CANONYM interface for establishment of privacy preferences.

$\sum a_o^j$ delivered in Alg. 4, and the Lagrange interpolation is finally computed, concluding the JRSS and reaching the jointed policy (Alg. 3).

We conducted experiments over different sizes for group of Co-owners to measure the average time of execution of the JRSS algorithm which is $\mathcal{O}(n)$ in terms of the communication cost between n entities. A Table *albumPreferences* stored at the *Policies DDBB* centralises communications and keeps traces of (i) the identifier of the album to which the joint policy should be applied (*idAlbum*); (ii) the identifier of the co-owner who defines a particular preference (*idUserOrigin*); (iii) the identifier of the co-owner applied for the polynomials calculations (*idUserDestination*); and (iv) $(t-1)$ random values $(a1, a2, a3)$ also related to polynomials calculations. Note that the value of the personal polynomial in each user's ID is locally stored to prevent the DDBB from inferring users preferences. This database is also inaccessible by Co-owner's client. This phase then linearly depends on the number of co-owners involved and their promptness of successfully completing the

Algorithm 3 Final polynomial

```
array x = idsUsers
array y
array valuesAllUsersAtt1 = valuePolyAtt1[i] AllUsers
array valuesAllUsersAtt2 = valuePolyAtt2[i] AllUsers
array valuesAllUsersAtt3 = valuePolyAtt3[i] AllUsers
for i=0:numAttInPolicy do
    y[i] = valuesAllUsersAtt+"i"
end for
array finalPolicy
for i=0:numAttInPolicy do
    finalPolicy[i]=interpolateLagrange(x,y[i])
end for
```

privacy setting. Computation at every node is negligible.

A Note on the Incentives for Co-operation Enforcement. This phase requires Co-owners to co-operate to be successfully completed. Existing approaches to stimulate and foster user/node cooperation are mainly based on the application of incentives mechanisms. For example, in [21] the use of a Clarke-Tax algorithm provides participating parties with the incentive to provide their privacy preferences in the collectively specification of the access policy. A tax will be assessed to each individual. The amount of this tax depends on how the individual's utilities affect the rest of the group. Policies are then established regarding the maximization of the social utility and ensuring the trustfulness of co-owners.

5.3 Resource Access Implementation and Evaluation

The access to a given photo album is initiated by a User client requesting it to the server side. At the client side, a *Credential Matcher* procedure queries the ABC_DDBB for a valid ABC to perform the credential proving and sends results to the Co-owner. Algorithm 4 and 5 depict the pseudocode, from the User and Provider point of view respectively, showing the steps needed to prove and verify that the user's private attributes (x_1, x_2, x_3) satisfy the boolean formula $\Phi_{\xi}(x_1, x_2, x_3) = 1$. Note that mathematical details follow [28] where similar details for the rest of proofs are fully demonstrated.

We also evaluate the computational[10] cost in terms of time spent by the *Resource Access* phase as the User has to create a randomized version of his

[10] Technical details: The analysis has been performed on a 3GHz Intel Core 2 Duo, 4Gb RAM.

Algorithm 4 ZKPoK1 commitment (Client-side credential proving)

```
vas x, x1, x3 g, g1, g2, g3 are public
var e = getE()
var N = getN()
var t = numNodes
var $\gamma$ = random().inGroup(Pow(2,t))
var $\alpha$ = valuePolyAtt1[0]
var $\beta$ = valuePolyAtt2[0]
var y = random().inGroup(e)
var rprime = random().inGroupCoprimes(N)
var ry = random().inGroup(e)
var r1 = random().inGroup(e)
var r2 = random().inGroup(e)
var tprimeAux=Pow(g,ry) * Pow((g1* Pow(g2, $\alpha$)),r1) * Pow(g3,r3)* Pow(rprime,e)
var tprime= Mod(tprimeAux, N) {Mod(x,y) means x mod y}
var sy=$\gamma$ * y + Mod(ry, e)
var s1=$\gamma$ * x1 + Mod(r1, e)
var s3=$\gamma$ * x3 + Mod(r3, e)
var sprimeAux=Pow(g, ($\gamma$ * y + ry)) * Pow((g1 * Pow(g1,$\alpha$)), ($\gamma$ * x1 + r1)) *
    Pow(g3, ($\gamma$ * x3 + r3)) * Pow(x, y) * rprime
var sprime= Mod(sprimeAux, N)
```

ABC upon each request of access to the protected content. Our prototype interface allows us to test the User client creating the credential token and the server manager verifying the policy. On average, we found that creating such credential takes 28143 ms whereas its verification takes only 8138 ms. Additionally, in the case of unsuccessful verifications when User's ABC does not satisfy the privacy policy, the overhead is reduced to an upper bound of 8272 ms. Also, note that data transmission must be anonymized so an extra delay has to be added. We believe that our system computational cost is more than acceptable as to manage data collections in current OSN applications.

Moreover, we can compare our system with closely related work such as the framework in [15] also implemented as a FB application. In their experiments, authors found linearity between the framework performance and the resource size, mainly due to the use of content encryption, e.g. accessing a 500KiB content took about 400000 ms. Our implementation does not suffer from such a data length dependency. The *Co-ownership Management* phase is affected, however, by the lack of synchronization of co-owners to provide their privacy preferences. Cooperation is then needed and its enforcement is out of the scope of this paper. To this regard, a simple analytical model based on Cooperative Game Theory is introduced

Algorithm 5 ZKPoK1 commitment verification (Co-owner-side credential proving)

```
Received vars: sy, s1, s3, sprime, tprime, $\gamma$ {Note that in this implementation, for
  being a simulation, contrary to [28] tprime, $\gamma$ are also sent to the Provider}
var ahatAux= Mod((Pow(g,y)*m), N)
var partVerification1 = Pow((ahat* (Pow(g2,-$\beta$))), $\gamma$) * tprime
var partVerification2Aux = Pow(g, sy) * Pow((g1 * Pow(g2, $\alpha$)), s1) * Pow(g3, s3) *
  Pow(sprime,e)
var partVerification2 = Mod(partVerification2Aux, N)
if partVerification1 = partVerification2 then
    Access granted if ZKPOK2, ZKPOK3 and ZKPOK4 are also correct
else
    Access denied
end if
```

in [29] to evaluate the feasibility and outcomes of cooperative interactions in OSNs, being this model applicable to this proposal's validation.

A Note on the Usability Evaluation. Usability does have an impact on privacy management in OSNs like FB. For this reason, our prototype is functional and has been built to measure the efficiency and effectiveness of the crypto functions used. The interface created for managing the new privacy settings is web-based and tries to minimise complex user interactions. Though the *Co-ownership Management* phase relies on the cooperation and (desired real-time) disposal of the co-owners, experimental results showed that the overhead incurred is acceptable. Our immediate work is to develop this functionality as a FB application which simulates photo management through the proposed technique to measure the real users' satisfaction.

6 Conclusion

In this paper we have proposed and implemented a privacy-enhanced access control system for OSNs which deals with both, co-ownership management of shared data and anonymous access to that data. The system's main application scenario is of sharing large collections of photos or privacy-aware data, involving multiple co-owners who keep their privacy preferences anonymous in the collectively construction of the data's access policy. Any system user can be grated access to the data iff the user attributes satisfy the policy. Besides, some remarkable strengths are noticed. First, privacy preferences are secretly defined by each co-owner and, neither other users nor the OSN site server are able to learn about them. Secondly, users can be authenticated and authorised anonymously preventing privacy related concerns like profiling and gossiping. Finally, the use of attributes-based access control policies facilitates a fine-grained access control management. Immediate future work is on the development of the proposed functionality as a FB plugin to evaluate real usability. Furthermore, the need of cooperation of co-owner at the *Co-ownership Management* phase can be alleviated by an incentive mechanism.

References

[1] Y. Liu, K. P. Gummadi, B. Krishnamurthy, A. Mislove, Analyzing facebook privacy settings: user expectations vs. reality, in: Procs. of the 2011 ACM SIGCOMM conf. on Internet measurement conference, 2011, pp. 61–70.

[2] M. H. Ibrahim, Efficient dealer-less threshold sharing of standard rsa, International Journal of Network Security 8 (2) (2009) 139–150.

[3] G. Persiano, I. Visconti, An efficient and usable multi-show non-transferable anonymous credential system, in: Financial Cryptography, Vol. 3110, 2004, pp. 196–211.

[4] M. Research, U-prove anonymous credential system based on brands' work, http://research.microsoft.com/en-us/projects/u-prove/.

[5] S. Petronio, Communication privacy management theory: What do we know about family privacy regulation?, Journal of Family Theory & Review 2 (3).

[6] A. Squicciarini, H. Xu, X. Zhang, Cope: Enabling collaborative privacy management in online social networks, Journal of the American Society for Information Science and Technology 62 (3) (2011) 521–534.

[7] H. Hu, G. Ahn, Multiparty authorization framework for data sharing in online social networks, in: Procs. of the 25th annual IFIP WG 11.3 conf. on Data and applications security and privacy, DBSec'11, 2011, pp. 29–43.

[8] H. Hu, G.-J. Ahn, J. Jorgensen, Multiparty access control for online social networks: model and mechanisms, Knowledge and Data Engineering, IEEE Transactions on 25 (7) (2013) 1614–1627.

[9] B. Carminati, E. Ferrari, Collaborative access control in on-line social networks, in: Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom), 2011 7th International Conference on, IEEE, 2011, pp. 231–240.

[10] K. Thomas, C. Grier, D. Nicol, unfriendly: multi-party privacy risks in social networks, in: Proceedings of the 10th international conference on Privacy enhancing technologies, PETS'10, 2010, pp. 236–252.

[11] L. González-Manzano, A. I. González-Tablas, J. M. de Fuentes, A. Ribagorda, Cooped: Co-owned personal data management, Computers & Security.

[12] J. Domingo-Ferrer, A. Viejo, F. Sebe, U. Gonzalez-Nicolas, Privacy homomorphisms for social networks with private relationships, Computer Networks 52 (15) (2008) 3007 – 3016.

[13] R. Sayaf, D. Clarke, Access control models for online social networks, Social Network Engineering for Secure Web Data and Services (2012) 32–65.

[14] R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin, Persona: an online social network with user-defined privacy, in: ACM SIGCOMM Computer Communication Review, Vol. 39, ACM, 2009, pp. 135–146.

[15] S. Braghin, V. Iovino, G. Persiano, A. Trombetta, Secure and policy-private resource sharing in an online social network, in: Procs. of the IEEE Int. Conf. on Social Computing, SocialCom/PASSAT, IEEE, 2011, pp. 872–875.

[16] A. Campan, T. M. Truta, Data and structural k-anonymity in social networks, in: F. Bonchi, E. Ferrari, W. Jiang, B. Malin (Eds.), Privacy, Security, and Trust in KDD, Springer-Verlag, 2009, pp. 33–54.

[17] M. Bertier, D. Frey, R. Guerraoui, A. Kermarrec, V. Leroy, The gossple anonymous social network, in: Procs. of the ACM/IFIP/USENIX 11th Int. Conf. on Middleware, 2010, pp. 191–211.

[18] P. M., M. W., N. B., Pisces: Anonymous communication using social networks, CoRR.

[19] N. Shang, M. Nabeel, F. Paci, E. Bertino, A privacy-preserving approach to policy-based content dissemination, in: Procs. of the IEEE 26th Int. Conf. on Data Engineering (ICDE), 2010, pp. 944–955.

[20] L.-H. Vu, K. Aberer, S. Buchegger, A. Datta, Enabling secure secret sharing in distributed online social networks, in: Computer Security Applications Conf., 2009, pp. 419–428.

[21] A. C. Squicciarini, M. Shehab, J. Wede, Privacy policies for shared content in social network sites, The VLDB Journal 19 (6) (2010) 777–796.

[22] S. Jahid, S. Nilizadeh, P. Mittal, N. Borisov, A. Kapadia, Decent: A decentralized architecture for enforcing privacy in online social networks, in: In Int. Conf. on Pervasive Computing and Communications Workshops, IEEE, 2012, pp. 326–332.

[23] S. Braghin, V. Iovino, G. Persiano, A. Trombetta, Secure and policy-private resource sharing in an online social network, in: In the 3rd int. conf. on social computing, IEEE, 2011, pp. 872–875.

[24] J. Park, R. Sandhu, Y. Cheng, A user-activity-centric framework for access control in online social networks, Internet Computing, IEEE 15 (5) (2011) 62–65.

[25] L.-H. Vu, K. Aberer, S. Buchegger, A. Datta, Enabling secure secret sharing in distributed online social networks, in: Procs. of the Annual Computer Security Applications Conf., 2009, pp. 419–428.

[26] S. A. Brands, Rethinking public key infrastructures and digital certificates; building in pivacy, in: MIT Press, 2000.

[27] J. Camenisch, E. Van Herreweghen, Design and implementation of the idemix anonymous credential system, in: Procs. of the 9th ACM conf. on Computer and communications security, CCS '02, 2002, pp. 21–30.

[28] A. Alcaide, E. Palomar, J. Montero-Castillo, A. Ribagorda, Anonymous authentication for privacy-preserving iot target-driven applications, Computers & Security 37 (2013) 111–123.

[29] E. Palomar, A. Alcaide, E. Molina, Y. Zhang, Coalitional games for the management of anonymous access in online social networks, in: Procs. of the 11th Annual Int. Conf. on Privacy, Security and Trust, PST, IEEE, 2013, pp. 1–10.