User-Managed Access Control in Web Based Social Networks

Lorena González-Manzano[1] Ana I. González-Tablas[2] José M. de Fuentes[3] Arturo Ribagorda[4]

[1] Avda. de la Universidad, 30. Computer Science and Engineering Department.
University Carlos III of Madrid, 28911 Leganés, Spain
lgmanzan@inf.uc3m.es

[2] Avda. de la Universidad, 30. Computer Science and Engineering Department.
University Carlos III of Madrid, 28911 Leganés, Spain
aigonzal@inf.uc3m.es

[3] Avda. de la Universidad, 30. Computer Science and Engineering Department.
University Carlos III of Madrid, 28911 Leganés, Spain
jfuentes@inf.uc3m.es

[4] Avda. de la Universidad, 30. Computer Science and Engineering Department.
University Carlos III of Madrid, 28911 Leganés, Spain
arturo@inf.uc3m.es

1 Introduction

Internet has become one of the main communication media, users as well as services have increased year after year[5]. Indeed, Web Based Social Networks (WBSNs) are one of the most recently ways of communication. According to [1], since the beginning of social networks in 1997 and their establishment with Friendster in 2002, many of them have appeared and their population has intensively increased. For example, three years after the emergence of Facebook there were more than 52 millions of registered users, currently achieving about 850 million[6]. Likewise, internet users in 2002 reached 587 millions and nowadays there is a population larger than 2095 millions[7]. As a result, simplifying and only considering Facebook, notice that 35,79% of internet users make use of this WBSN. This matter leads us to wonder about the huge importance of analyzing and improving all kind of features of this type of systems.

By contrast, given the huge quantity of users and data managed, a key issue comes into place, privacy. This feature is defined as "the condition of not having undocumented personal knowledge about one possessed by other" [2]. WBSNs are systems that store a great amount of personal information

[5] http://www.internetworldstats.com/emarketing.htm, last access November 2012

[6] http://www.internetworldstats.com/facebook.htm, last access November 2012

[7] http://www.internetworldstats.com/stats.htm, last access November 2012

that must be carefully protected, even been an issue not deeply recognized and taken into account by users. For instance, J. Becker *et al.* studied that the total of Facebook users have never used any of the privacy mechanisms provided [3]. Likewise, Acquisti *et al.* analyzed that even Facebook users who are aware of privacy problems continue using it [4]. Namely, this matter may be related to the enormous appearance of perceived benefits, as well as, to the fact that people may be conscious about internet security but not aware of threats [5]. By contrast, a much more recent study identifies that WBSN users, in regard to New York city Facebook users, have become much more private [6].

Despite interests and motivations of people, privacy is extremely relevant in everybody's life and recently it has been extensively studied and defined. For instance, in the Universal Declaration of Human Rights it is set the right to not have interferences with our privacy[8]. In fact, although users are not specially focused and worried about privacy issues, the chief point is that laws and rights highlight the necessity of protecting everybody's privacy. Furthermore, as C. Dwyer *et al.* pointed out, WBSNs need to provide enough data protection mechanisms to achieve the same level of privacy found offline [7].

Considering the importance of privacy, previously mentioned, together with the increase of WBSN users, a crucial question arises: Do WBSNs provide enough mechanisms to preserve privacy? This question has not got a simple answer, as even a previous question is still not clear: Which requirements must be fulfilled to appropriately protect users privacy in WBSNs? In 2007, Gates identified a set of requirements in order to provide user-managed access control in WBSNs [8] which remains that a proper access control management contributes to privacy preservation. These requirements establish that access control systems developed for WBSN must be *relationship-based*, *fine-grained*, *interoperable* and follow the *sticky-policy* paradigm. In this work, Gates' requirement list is taken as a starting point and a new requirement, *data exposure minimization*, is identified in the light of recent trends in currently active WBSNs [9]. Since then, several research works have proposed access control systems that address some of the requirements identified by Gates, but not all of them or partially. Moreover, there is not a clear path that guides researchers and industry in the task of developing access control systems for WBSNs that address the whole set of requirements.

There are some contributions that try to analyze security in WBSNs. R. Ajami *et al.* [10] study security challenges of WBSNs, focusing on identifying privacy, anonymity and security risks, as well as, describing some security WBSN proposals, such as VIS or FlyByNight (also analyzed in this work). Likewise, in [9] breaches and security mechanisms for WBSNs are identified. Moreover, a set of possible attacks like information leakage, de-anonymization and phising are noticed. From a similar point of view but based on WBSNs privacy, E. Zheleva and L. Getoor present a description of privacy breaches and

[8] http://www.un.org/en/index.shtml, last access November 2012

attacks [11]. Nonetheless, regarding purposes of this work, the most similar approach is [12]. It focuses on access control in WBSNs, specifying requirements that, contrary to the user centric perspective described herein, are based on access control policy languages and access control mechanisms.

Therefore, identified problems are twofold, (1) the lack of a model that particularly addresses the whole set of requirements and (2) the lack of specific mechanisms to manage them. Due to this fact, this paper presents three main contributions. The first one is the proposal of $SoNeUCON_{ABC}$, an access control model that allows the fulfillment (from a theoretical point of view) of the whole set of requirements as an extension of the $UCON_{ABC}$ usage control model [13], and a basic mechanism that implements it. The second contribution of this work is to select a set of mechanisms recently proposed in the literature that allow fulfilling the remaining three requirements on top of $SoNeUCON_{ABC}$. The third contribution is to analyze in detail to which degree recent academic proposals for WBSNs and currently deployed WBSNs satisfy the identified requirements or adopt any of the mechanisms that would facilitate the satisfaction of any of them. In this sense, 25 academic proposals and 9 WBSNs in use are analyzed.

The rest of the paper is structured as follows. Section 2 contains a brief background on access control models and mechanisms. Section 3 provides a conceptualization of a WBSN, necessary to develop the contributions of this work. Then, Section 4 details the requirements identified by Gates in [8] and the new one added herein. In Section 5 the proposed access control model, $SoNeUCON_{ABC}$, and the specific mechanism selected to fulfill the whole set of requirements is presented. Next, in Section 6, the results of the analysis of WBSNs regarding the satisfaction of the set of user-managed access control requirements are summarized. Finally, Section 7 contains conclusions and open research issues.

2 Background

In this Section, an overview of traditional and recent access control models is provided, as well as a brief depiction of the mechanisms that implement these models.

2.1 Access control models

Currently three traditional access control models can be identified: Mandatory Access Control (MAC), in which objects and subjects are classified according to security levels and access is granted in regard to them; Discretionary Access Control (DAC), in which access to information is carried out in respect to the user's identity and a set of authorizations or rules; and Role Based Access Control (RBAC), based on the definition of different roles, assigning permissions to roles, and, then, assigning roles to subjects [14]. More specifically, RBAC has been extended in other models: TrustBAC [15], based on trust instead of roles; TBAC [15], focused on dynamically managing permissions once tasks completeness; TRBAC [16], based on activating or deactivating a role in regard to time specifications; RelBAC [15, 17, 18] in which permissions are modeled as relationships between users and data while access policies are instances of them; LRBAC [19] that adds location constraints in the traditional approach; or GRBAC [20] that focuses on including environment and object roles to the basic approach in which only subject roles are tackled.

However, another new type of access control model has recently appeared, Attribute Based Access Control (ABAC) [21, 22] that is focused on the definition of policies considering attributes of subjects, resources and the environment. Indeed, under this perspective, the Usage Control Model ($UCON_{ABC}$) [13, 23], extensively developed by J. Park and R. Sandhu, has been carried out. It focuses on presenting an unified framework for access control in which MAC, DAC, RBAC and DRM (Digital Rights Management) systems are likely to belong to.

More specifically, the $UCON_{ABC}$ model considers eight components: *subjects (S)*, that are entities that exercise rights on objects; *objects (O)*, that are entities which subjects hold the right on; *subject attributes (ATT(S))* and *object attributes (ATT(O))* that refer to features associated with subjects and objects, respectively; *rights (R)*, which are recognized as privileges exercised on objects such as read or write; *Authorizations (A)*, that correspond to predicates on subject and object attributes that are evaluated in order to decide whether the requested right on a specific object made by a certain subject should be allowed or denied; *oBligations (B)*, that represent predicates that must be satisfied before, during or after the right is granted; and *Conditions (C)*, that correspond to environmental or system factors not controlled by subjects and which are taken into account during the access decision process.

In $UCON_{ABC}$, *usage decision functions* are pointed out in order to permit or deny access. They are based on authorizations, obligations and conditions, but two other factors are also considered: the *mutability of attributes* and the *continuity of decision*. On the one hand, mutability refers to the possibility of updating subject and object attributes at different times in respect to the moment at which the right is exercised, namely, before (pre), during (ongoing) or after (post), besides considering inmutable attributes. On the other hand, continuity of decision refers to the possibility of persistently verifying policies while the right is exercised, which is referred to as on-going decision; the other possibility is checking policies only before the right is exercised, which is referred to as pre-decision.

In the original $UCON_{ABC}$ model, it is assumed that an access control policy is defined by the system's administrator and this policy is applied to all users in the system. A recent work by Salim *et al.* propose an administrative model, orthogonal to the $UCON_{ABC}$ model, where the attributes and rights of subjects and objects are established through assertions made by authorized subjects [24]. Specifically, this administrative model allows the specification of attributes and rights by the entity that has the authority to establish them.

2.2 Access control mechanisms

Access control models are implemented through access control mechanisms. The most traditional ones are access control lists (ACL) and capabilities. ACLs inform about the authorized permissions in regard to a single resource and multiple users. On the contrary, capabilities attest the permissions granted to a single user for multiple resources.

Every access control system has an associated architecture. Usually, the core component is a reference monitor that consists of two elements, namely the Policy Decision Point (PDP) and the Policy Enforcement Point (PEP)[9]. The former provides affirmative or negative responses in regard to the requested rights over objects according to the defined policies. The latter enforces decisions taken by the PDP. Both elements are considered trusted entities.

Several architectures may be adopted depending on the number and location of the PDP and PEP, and the nature of the interactions among the parties involved in an access request[9]. Depending of the location of the PDP and PEP, architectures can be classified among server, client or client-and-server side. In the first case, the reference monitor is deployed in the server, in the second case, in the client, and in the third, the components of the reference monitor are deployed in both locations. Regarding the number of components involved in the reference monitor, different n-tier architectures can be identified. For example, a two-tier architecture may consider a server-side reference monitor combined with a client-side reference monitor.

The $UCON_{ABC}$ model allows its implementation with several architectures depending of the concrete features selected from the model. If ongoing decision is required, the PDP and PEP need to be permanently interacting with each other and state-full. Thus, this continuous connection links both to attribute mutability and privilege revocation. In case any security policy is violated during a resource usage, the PDP communicates with the PEP in order to enforce the end of the usage process.

Apart from ACLs and capabilities, a third access control mechanism is referred to as ‘lock and key’. In its cryptographic implementation, data is stored encrypted (locked) and it can only be disclosed if the appropriate description key is known [25]. This type of mechanism is recently attracting a lot of attention from data protection researchers. For instance, Personal Data Servers (PDS) are storages of encrypted personal data that intend to minimize the consequences of having compromised the storage device [26, 27, 28, 29]. In order to access to data, keys must be previously exchanged between contestants. If access control is cryptographically enforced, no reference monitor is required [30].

[9] http://tools.ietf.org/html/rfc2904, last access November 2012

Additionally, there are other recent approaches which encrypt data in respect to attributes, known as Attribute Based Encryption (ABE) schemes. These proposals focus on creating a pair of keys, to encrypt and decrypt, in regard to an established group of attributes. Indeed, there are many ways of performing it, either using a single Central Authority (CA) to generate and distribute keys [31, 32, 33, 34], or using some CAs to decentralize key management [33, 34] or even developing a protocol to remove the necessity of a CA [35, 36]. ABE schemes can be divided into two groups, Ciphertext-Policy ABE (CP-ABE) [31, 34] and Key-Policy ABE (KP-ABE) [32, 33, 35, 36]. The former corresponds to the association of policies with ciphertexts and attributes to user keys and it is a remarkable technique in applications in which data is managed by multiple profiles, such as in hospitals or in the army. By contrast, the latter corresponds to the attachment of policies to user keys and attributes to ciphertexts, being useful in applications like auditing logs. The main difference between both approaches is that in CP-ABE attributes of key users are known, while in KP-ABE they are hidden. Besides, it is noticeable that both techniques have some limitation [37]: it is essential the existence of an authority to provide keys and flexibility of access control policy definition is currently restricted because disjunctions and conjunctions are the only operators used. As a result, much more work is required in spite of having appeared proposals to handle some of these problems [35, 36, 38].

Finally, revocation is a challenging feature that has to be managed in any access control mechanism. Specifically, it refers to the modification of access control policies. Some techniques are relatively simple such as using a revocation list [39, 40] or expiration times [41] while others are more sophisticated, for example the use of a PKI [42] or a particular scheme [43]. Besides, revocation is related to attribute modification. Once an attribute is modified the most appropriate procedure is to re-evaluate policies and act accordingly.

3 Conceptualization of WBSNs

WBSNs allow their users to establish relationships each other and share their data. In this Section, a conceptualization of WBSNs is presented to lay the bases for understanding the proposal and its later analysis. Commonly, WBSNs are modelled as graphs, being Harary who, in 1953, applied graph theory regarding group behaviour, social pressure, cooperation, power and leadership [44]. Indeed, Harary is considered the pioneering of the application of graph theory to network analysis.

More specifically, a graph is characterized by a huge quantity of entities, called nodes, and a vast quantity of connections between the nodes, called edges. In general terms, when modelling a WBSN as a graph, users correspond to the nodes and users relationships to the edges. This type of representation has been used by many authors in recent literature [45, 46, 47, 48], being Carminati one of the most representative.

3.1 Data

In a WBSN, the set of considered data types may include photos, videos, wall messages and personal messages that are private and directly written to a certain person or a group of people. Furthermore, according to Bruce Schneier, data in WBSNs can be further classified as: *service data*, data you give to the WBSN for using it; *disclosed data*, data posted in your own pages; *entrusted data*, data you post on other user's pages; *incidental data*, data other users post about you; *behavioral data*, data about your habits that sites collect; and *derived data*, data about yourself derived from what other users say [49]. In the conceptualization presented herein, only *service*, *disclosed* and *entrusted data* is considered since the rest of data types require a great deal of management complexity and such management is, indeed, far from being satisfactorily solved yet. The set containing all data will be referred to as D.

Additionally, data has a set of attributes associated to it, such that $dAT = \{dat_1, dat_2, ..., dat_{n_{dAT}}\}$ where n_{vAT} is the total number of data attributes. Data attributes can be classified in two groups. First group involves own features of data such as type of data, creation time, size and so on. Second group refers to any kind of characteristic that can be assigned to data, for example the fact of being private, confidential or public, or the topic of the data among others.

3.2 Actions

In a WBSN several actions can be performed on data. The set of defined actions is denoted as AC. Main four actions that can be performed over data are: *read*, equivalent to visualize any kind of content; *update*, equivalent to write down tags in videos or photos, or changing any commentary previously written; *insert* an element, equivalent to upload a photo or a video to the WBSN; and *delete* an element. Nonetheless, if needed, more actions can be considered.

3.3 Users

The set of users of a WBSN is identified herein as V. In [50] users are classified in three groups: *owners*, *originators* and *viewers*. Originators are those users that originally create and upload a specific data to the WBSN. Owners, who can be a single user or a group of users, administer access to the data they own and share ownership privileges with the data originator. Finally, viewers are those users that request access to a certain data. However, it is noticed that conflicts between owners and originators are a matter of vital concern regarding privacy in WBSNs. For example, if a user A takes a picture of a user B and uploads it to the data space of user C in the WBSN, who has ownership rights over the picture? In fact, the right question is not who has ownership rights, but who has administrator rights over the picture [51].

This issue is simplified in current WBSNs, such as the management of wall data in Facebook where it is established that if a user A writes in the wall of user B, B becomes the owner of the message. Another example where this problem is illustrated is that of a parent supervising access control and privacy preferences of his/her child in a WBSN. The parent is not the owner nor the originator of the data uploaded by the child to the WBSN, but he should be allowed to co-administer the access control policies of his child. This matter must be carefully studied and analyzed in future work; furthermore, it will be discussed in Section 7. Therefore, herein, for simplicity and without losing generality and considering a single piece of data, d_i, it is distinguished the data *administrator*, i.e., $administrator(d_i)$, that is the user who administers the data access controls, and the data *requesters*, i.e., the remaining users of the WBSN that request access to that piece of data. Therefore, all the data administered by a user v_i is denoted as D_i, where $D_i \subseteq D$. If a single administrator is considered for each piece of data, it follows that $D = \bigcup_i D_i$ and $\forall\ i, j\ D_i \cap D_j = \emptyset$.

As it happened with data, users may have also a set of associated attributes. This set of attributes is referenced as $vAT = \{vat_1, vat_2, ..., vat_{n_{vAT}}\}$ where n_{vAT} is the total number of users attributes. Some of these attributes can be reflected as data in the WBSN. On the one hand, a user profile links each user with his/her nationality, age, music preferences and so on. On the other hand, there is other group of attributes, called user contextual attributes [52], that describe a user's personal mood, like happy, nervous and so on, or his/her current activity like eating, running, etc. A user's location is a particularly relevant user contextual attribute in WBSNs like Google Latitude, in which this information is the main data managed, or in WBSNs such as Facebook Places, in which user's location can be associated to data.

3.4 Relationships

In a WBSN, a user can usually accept and withdraw the establishment of relationships with other users of the WBSN. As previously identified, WBSN relationships corresponds to edges of the social graph which connect directly or indirectly pairs of users.

Direct relationships between users of a WBSN are identified herein as E. A set of attributes can be associated to a direct relationship. This set is denoted as $eAT_i = \{eat_1, eat_2, ..., eat_{n_{eAT}}\}$. The most important of these attributes are the relationship direction and the relationship type. The former refers to the unidirectional or bidirectional nature of the relationships and the latter represents the relationship semantic meaning, which is also called the relationship role by some authors.

On the one hand, regarding relationship direction, relationships can be unidirectional or bidirectional [47]. The former corresponds to relationships in which a relationship request is only established in one direction. For example, given users A and B, if A makes a friend request to B and B accepts the request, then, A is said to be friend of B but it cannot be said the same in the other way round (i.e., B cannot be said to be friend of A). On the contrary, the latter, a bidirectional relationship, implies that both nodes associated to a relationship request, once accepted, have the same type of relationship in both directions. Specifically, bidirectional relationships can be identified as a pair of unidirectional relationships in which both nodes are issuers and recipients of the relationships at the same time. Indeed, this is the type used in current social networks.

On the other hand, semantic meaning or role of relationships imply having different relationships such as ‘friend’, ‘professional’, ‘family’ and so on. Some social networks focus on a specific role, for example LinkedIn is based on professional relationships. On the contrary, there are social networks in which different roles can be considered.

For example, in Facebook there exists the possibility of managing groups of people according to established preferences, e.g., a group of ‘family’, a group of ‘bestFriends’ and so on. Furthermore, multiple direct relationships may be considered between two nodes, each one with a different role.

Relationships may have other attributes such as creation time or history. Additionally, users may attach other attributes to their relationships such as a level of trust [47] and certain duration. Level of trust corresponds to a quantitative measure to determine the strength or weakness of a relationship. Concerning duration, it is the time during which the relationship remains valid.

Two users of a WBSN may not be directly connected but indirectly, that is, a direct relationship does not exist between them but a path, P, connecting both nodes can be found considering other users and their relationships. The path consists of an ordered set of edges or direct relationships such that $P = \{e_1, e_2, ..., e_n\}$. In particular, a relationship between two users v_i and v_j is denoted as P_{v_i,v_j}. The number of edges in the path is referred to the length of the relationship. Note that if length equals one, it is the case of a direct relationship, and indirect if length is greater than one [53].

Therefore, the concept of user relationships and their attributes can be generalized to consider both direct and indirect types. The set of such relationships is then denoted as P and their attributes $pAT = \{pAT_1, pAT_2, ..., pAT_{n_{pAT}}\}$, where $pAT_i = \{eAT_1, eAT_2, ..., eAT_n\}$. These attributes are similar to those associated to direct user relationships and can be derived from the attributes of the set of edges that compose the path. That is, given attributes $pAT_j \ \in \ pAT$ and a path $p_k \ \in P$, each $eat_n \in eAT_n \in pAT_n$ is calculated as $eat_i(p_k) \ = f_{\{\ i\ =\ 1,\ length(p_k)\}}(e_i)$, where e_i are the edges that compose p_k. For example, given three users, v_1, v_2 and v_3, connected in pairs by $e_1 = \{v_1, v_2\}$ and $e_2 = \{v_2, v_3\}$, the path between v_1 and v_3 is $p_1 = \{e_1, e_2\}$. As proposed by Carminati *et al.* in [46], the level of trust of such relationship p_1 can be calculated as $trust(p_1) = \sum_{\ i\ =\ 1,\ length(p_1)} trust(e_i)$. The set of functions used to derive the value of the attributes $pat \ \in \ pAT$ will be denoted by F and its definition is far from being a minor issue.

3.5 Context

Context information is other aspect that may be considered when modeling a WBSN. The set of these features is denoted as *CX*. Dynamic, assorted and external features such as communication network status, service availability or other quality parameters can be involved here.

3.6 Summary of the conceptualization

The conceptualization provided above is summarized in this Section and depicted in Figure 1.

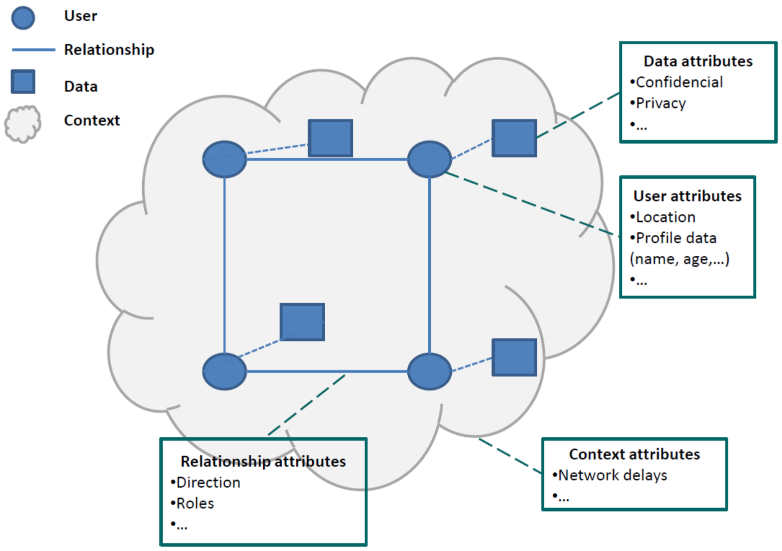


Fig. 1. Web Based Social Network Conceptualization

Thus, a WBSN can be conceptualized as: WBSN = {V, vAT, P, pAT, F, D, dAT, AC, CX}. V corresponds to the graph nodes, which represent the WBSN users. P are the graph edges which represent the relationships which connect two users. D represents data and AC corresponds to actions that can be performed on data. vAT, dAT and pAT represent the set of user, data and relationship attributes, respectively, and F the set of functions that allow the derivation of attributes values associated with indirect relationships. Finally, CX represents the system's context.

In particular, eAT, given that $eAT \in pAT$, considers, summing up, direction, role, level of trust, duration and own features such as history or creation time; vAT mainly corresponds to profile data and location; and dAT are features assigned such as confidentiality or privacy labels, related topic, and other own features like creation time.

Note that, in the provided conceptualization, it is assumed that the set of user, data and relationship attributes is defined by the system, and that their value can be assigned by one of the involved parties. For example, the WBSN can assign to a piece of data its creation time, type, etc.; a user can assign the type, level of trust and duration to one of his/her relationship; external entities can be in charge of attesting real user attributes such as name, age, or driving license.

Finally, notice that as the conceptualization presented above is performed from a logical point of view, the way in which the storage is carried out, centralized like Facebook, decentralized such as Diaspora, encrypted or in plaintext, as well as, the way of specifying and managing relationships are not particularly considered.

4 Requirements for user-managed access control in WBSNs

In [8] a set of requirements are identified as key in order to successfully develop user-managed access control in Web 2.0, which includes WBSNs. These requirements are described next.

1. *relationship-based*: data administrators (e.g., the data owner) control the release of data based on the established relationships with data requesters, instead of delivering information depending on the requester role or any other feature.
2. *fine-grained*: users must control their information, choosing who is able to access it and under which circumstances in a fine-grained way. It should be possible to define fine-grained policies for both data (a specific portion of data within a data structure) and requesters.
3. *interoperability*: users access multiple WBSNs and want that their data is used in a similar way in many of them. Access control systems should be interoperable between different WBSNs, so, it would be possible that user preferences follow the user whatever WBSN is used to access the user's data.
4. *sticky policies*: policies should follow the data to which they apply, preventing from uncontrolled data disclosures after being released.

In this work, a fifth desirable requirement is identified, *data exposure minimization* against honest-but-curious storage services. Servers may protect information stored against possible threats but users are unaware of the servers procedures and techniques to carry out this issue. Indeed, personal information is completely susceptible of being used by companies, specially advertising ones, to improve and develop their products according to users perceptions [40, 42]. Therefore, it is desirable that users have control over their personal information without letting servers access the user's data unknowingly.

5 Proposal for user-managed access control in WBSNs

One of the purposes of this work is to explore access control solutions that satisfy the five requirements for WBSN described in Section 4. Alluding to the identified problems, which refer to the necessity of a model that satisfies the whole set of requirements and the definition of the related mechanisms, the main pair of contributions are described next:

- First, a model that, from a theoretical perspective, satisfies the whole set of requirements and a mechanism that, in practice, satisfies the requirements of *relationship-based* and *fine-grained* are presented. The model is an extension of the $UCON_{ABC}$ usage control model proposed by Park *et al.* [54] and of the $UCON_{ABC}$ administrative model proposed by Salim *et al.* [24]. The model is recognized as $SoNeUCON_{ABC}$, from Social Network $UCON_{ABC}$. The $SoNeUCON_{ABC}$ model is presented in Section 5.1 and the basic mechanism that implements it, as well as the architecture attached to it, are described in Section 5.2.
- Second, mechanisms that can be used on the previous basic system to satisfy the other three requirements (i.e., *interoperability*, *sticky policies* and *data exposure minimization*) are discussed and the possibilities of integrating these mechanisms on the basic access control system are also noticed. Moreover, the particular architecture attached to them are described. Sections 5.3, 5.4 and 5.5 address each of the mentioned requirements respectively.

5.1 Extending the $UCON_{ABC}$ model to consider relationships

At first sight, in order to address the *relationship-based* requirement, the RelBAC approach [15, 17, 18] (mentioned in Section 2) can be pointed out as the ideal proposal for social networks. However, WBSNs, though one of its main features is being relationship-based, need access control systems that are also capable of considering fine-grained policies in regard to attributes of data and requesters. For example, geo-social networks consider location, which is a feature directly related to users and difficult to handle through relationships.

On the other hand, access control systems that follow the ABAC approach consider attributes of users, data and (not always) context as the main policy elements to take the decision about whether a requested action is authorized or not. Indeed, the ABAC approach allows fine-grained access control but on the negative side, it sets aside relationship management. Therefore, in this work, the $UCON_{ABC}$ model is selected as it is the most representative and mature ABAC-based model in comparison with other approaches like [55] or [56].

On the one hand, in [55], subjects, resources and contextual attributes are considered but other relevant elements such as rights or authorizations managed in $UCON_{ABC}$ are left aside. On the other hand, in [56], attributes are linked to the related entities without specifying particular types of them.

Under the assumption that the $UCON_{ABC}$ model is used, it is extended to include an exhaustive relationships management. Relationships can be included in the authorization decision process in different ways. For instance, relationships can be defined as an attribute of the origin entity of the relationship, i.e., a user Alice has a list of 'friends', a list of 'relatives', a list of 'co-workers', etc., and these lists are attributes of the user Alice. This is the approach taken in some recent proposals for WBSN that build on $UCON_{ABC}$ [57, 58]. Even the decentralized administrative model of Salim *et al.*, which has a direct application in WBSNs, takes this approach [24].

Nonetheless, WBSNs require the management of attributes as well as direct and indirect relationships. In this regard, even being an ongoing work, the study of WBSN features in respect to relationships management, at a starting point $UCON_{ABC}$ lacks indirect relationship management. To verify this issue Example 1 presents an access control policy that is tried to be defined by $UCON_{ABC}$.

Example 1

A user grants read access to users who are over 18 and who are friends of a friend of a friend.

According to $UCON_{ABC}$ the following subjects attributes ($ATT(S)$), objects attributes ($ATT(O)$) and predicates are required to define Example 1 access control policy:

- $ATT(S)$={Age, Friends} where Friends is the list of friends that the user has. Moreover, each list is composed of a set of attributes:
  - Friends=$\{User1_{Id}, relationshipTrust, ...\}$
- $ATT(O)$=$\{Object1_{Id}, ...\}$

- permit($s \in S,\ o \in O,\ r$): it grants access permission (r) over an object (o) to a particular subject (s).
- in($s \in S,\ Friends\ of\ v \in S$): it returns the existence of not of a friendship relationship between a pair of users (s and v). Notice that s has to be within the list of friends of v.

Then, considering that the administrator of the requested object o is referred to as a and s corresponds to the requester, the access control policy proposed in Example 1 and established by a is defined following [59]:

- $in(a, s.Friends)\ \wedge\ s.Age > 18\ \longrightarrow permit(s, a.o, r)$

At first, it can be thought that an assorted set of predicates can be devised to construct policies but their specification is restricted regarding elements involved in $UCON_{ABC}$. As pointed out in [57, 58], $UCON_{ABC}$ addresses relationships through subjects attributes and thus, they facilitate the management of direct relationships but prevent from managing the indirect ones. Consequently, a novel model, called $SoNeUCON_{ABC}$, is developed to address this issue. Nevertheless, a comparison and a deep analysis of relationships management in respect to existing access control models is a matter of future work (Section 7).

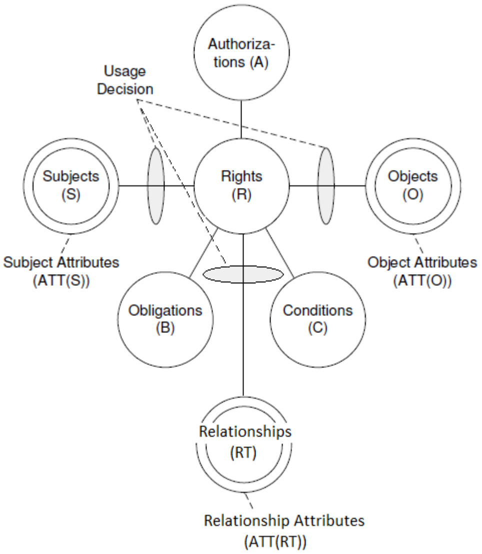


Fig. 2. $SoNeUCON_{ABC}$

Therefore, the $SoNeUCON_{ABC}$ access control model, an extension of the $UCON_{ABC}$ model is proposed in this work (Figure 2). According to $UCON_{ABC}$ a new independent entity, *relationship* (RT), and set of attributes, *relationship attributes* ($ATT(RT)$), are included. Note that either direct or indirect relationships are managed through attributes and, for instance, the attribute *length*, the distance between two nodes, can be applied to deal with the indirect ones. The point is that access control is managed through the establishment of policies in which $ATT(S)$, $ATT(O)$, $ATT(RT)$ and R are involved. The set of original entities, sets of attributes and functions considered in the $UCON_{ABC}$ model are also considered in the $SoNeUCON_{ABC}$ model. The elements of the WBSN conceptualization can then be related to those in the $SoNeUCON_{ABC}$ model:

- *Subjects* (S) are the WBSN users (V), previously identified as administrators and requesters; additionally, $ATT(S) \subseteq vAT$.
- *Objects* (O) are WBSN data (D), identified as photos, videos, wall messages and personal messages; additionally,
  $ATT(O) \subseteq dAT$.

- *Relationships* (RT) represent the set of relations that exist between a pair of users of the WBSN. Given a relationship between v_i and v_j, such set is noted as P(v_i,v_j). This set is composed of two subsets, namely forwards paths set (P"(v_i,v_j)) and backward paths set (P'(v_i,v_j)), such that P(v_i,v_j)= P"(v_i,v_j) ∪ P'(v_i,v_j). Both subsets contain different paths $p_k(v_i, v_j)$, where a path is a collection of direct relationships that form together a way to connect v_i and v_j.
  As an example let us calculate P(v_5,v_1) in the social network scenario depicted in Figure 3, where v_1 is the requester and v_5 is the administrator. In such network, all direct relationships are bidirectional. Thus, given a direct relationship between v_1 (the requester) and v_2 (the administrator), the forward edge is referred to as $e_{2,1}$ whereas the backward one is noted as $e_{1,2}$. The forward paths set P"(v_5,v_1) is given by: $P"(v_5, v_1) = \{\{e_{5,2}, e_{2,1}\}, \{e_{5,4}, e_{4,3}, e_{3,1}\}\}$.
  On the other hand, the backward paths set P'(v_5,v_1) is: $P'(v_5, v_1) = \{\{e_{2,5}, e_{1,2}\}, \{e_{4,5}, e_{3,4}, e_{1,3}\}\}$.
  Taking into account both subsets, the set P(v_5,v_1) is formed by their union, $P(v_5, v_1) = \{\{e_{5,2}, e_{2,1}\}, \{e_{5,4}, e_{4,3}, e_{3,1}\}\} \cup \{\{e_{2,5}, e_{1,2}\}, \{e_{4,5}, e_{3,4}, e_{1,3}\}\}$
  Finally, notice that $ATT(RT) \subseteq pAT$.

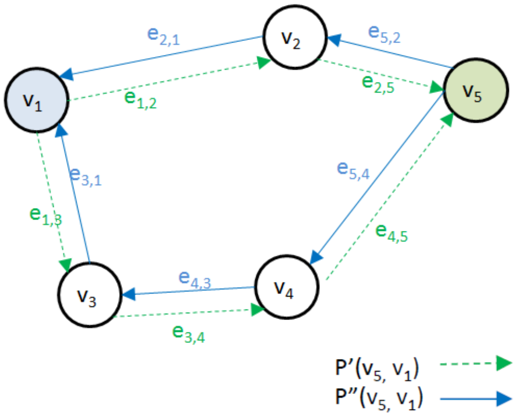


Fig. 3. Relationships example, P(v_5,v_1)

- *Rights* (R) refer to the actions (AC) that can be performed over WBSN data such as read, update or delete;
- *Authorizations* (A) are the rules defined as functional predicates that have to be satisfied in order to grant a subject a right on an object. Along this paper these elements will be called policies;
- *Obligations* (B) refers to activities that have to be carried out by the user before or while the usage process.

- *Conditions* (C) correspond to the previously identified set of context features (CX), such as network availability, etc.

Bearing in mind that the formalization of the model is not the goal of this paper, but a future open research issue, the definition of policies is briefly detailed. In particular, in the construction of policies (ρ) it is specified that S can perform R on O in respect to $ATT(S)$, $ATT(O)$ and $ATT(RT)$.

Each policy ρ in $SoNeUCON_{ABC}$ model is generally depicted as follows:

- $\rho(\rho_s; \rho_o; \rho_{rt}; r)$
  - $\rho_s = \{subject\ predicates\}$
  - $\rho_o = \{object\ predicates\}$
  - $\rho_{rt} = \{length\ \{\{path_1\}, \{path_2\}, ..., \{path_n\}\}\}$ where it is defined as a set of paths composed of a set of direct relationships, together with the use of the mandatory attribute *length* that corresponds to the distance between a pair of nodes (the administrator and the requester). It takes value $k \in \aleph$ and $1 \leqslant k \leqslant |S|$ because the maximum length is the total number of WBSNs users and also, it can take value "*" to express that all kind of relationships lengths are accepted. Moreover, paths can be described as unidirectional or bidirectional.
- $r = \{read \mid write \mid ...\}$

More specifically, the requested $r \in R$ over $o \in O$ is granted if $ATT(S)$ of the requester and $ATT(RT)$ between the requester and administrator satisfy the appropriate ρ. Notice that the request consists of the requester (s), the requested object (o) and the requested right (r) over the object, $Req = \{s, r, o\}$:

- $allowed(s, o, r) \Rightarrow \rho(\rho_s; \rho_o; \rho_{rt}; r)$

Finally, to have a better understanding of the creation of access control policies in $SoNeUCON_{ABC}$, the policy presented in Example 1 is defined:

- ρ=(age $>$ 18, $\emptyset$, {length=3 {role=friend},{role=friend},{role=friend}}, r), where $\emptyset$ implies that any particular set of restrictions is specified in relation to requested objects.

As a result, though requiring a detail definition, this paper presents $SoNeUCON_{ABC}$, a powerful access control model developed as an extension of the $UCON_{ABC}$ model to reach relationships management, involving both direct and indirect relationships.

5.2 Basic mechanism

Relationship-based and *fine-grained* requirements are already satisfied if the $SoNeUCON_{ABC}$ model is followed. On the one hand, the management of relationships is performed through the establishment of paths between users as it is described in Section 5.1. On the other hand, fine-grained management is carried out by the appropriate use of subjects, objects and relationship attributes, reaching the creation of assorted and expressive policies. Thus, the basic approach focuses on both requirements. However, there exist several possibilities for implementing it. In this approach, a server-side architecture that considers a single domain is proposed. This approach is depicted in Figure 4(a). Both data and policies are located in a single domain corresponding to the WBSN. Thus, the WBSN takes the role of both the PEP and PDP elements of the reference monitor. Data administrators or system administrators establish usage or administrative policies, and according them, the WBSN grants or denies requested access.

Nonetheless a significant point is the construction and management of access control policies. Regarding policies constructions, they could be defined according to all elements of the $SoNeUCON_{ABC}$ model, but the discussion in this work will be restricted to the authorizations element, as the issues related to the obligations and conditions elements do not present specific differences with respect to the original $UCON_{ABC}$ model and its implementations. Therefore, policies can be developed using any kind of user, relationship or data attribute, and actions. Note, however, that regarding the duration relationship attribute, it corresponds to an absolute time range during which the relationship exists. Analogous to a public key digital certificate, it can be described as "‘valid-from ..., valid-until..."’; for example, "‘valid-from 10th September 2011 12:00, valid-until 10th September 2011 14:00"’.

On the other hand, according to policies management, this architecture requires that administrators establish policies in the storage service (the WBSN) in respect to their access preferences. Then, in case that a change in the policies is needed (e.g. caused by revocation), they simply update the policies and the changes will be taken into account in the next access request. Note that this issue is, whilst different, related to the continuity of decision functions that may be used if mutable attributes are considered.

Finally, a challenging issue is the simplicity of the mechanism either for administrators or users. Administrators only create, establish and modify policies attached to data and requesters exclusively send requests to the PEP. However, there are a pair of relevant drawbacks. Firstly, as *data exposure minimization* requirement is not considered in this mechanism, data is not stored under user control and can be compromised. Then, trust in the storage device has to be assumed. Secondly, each time a user wishes to access to a specific data, policies must be satisfied. By doing this, policy enforcement is frequently performed and it can produce a bottleneck in the storage device (performance/workload).

5.3 Addressing interoperability

WBSNs are independent services that have different (although potentially similar) access control models, with different privacy policy definition languages and specific types of data and actions particular for each WBSN. Currently, small interactions are allowed between different WBSNs, such as importing a list of contacts or propagating messages in other WBSNs after they have been published in one of them. Naturally, users belonging to different WBSNs would like to have the possibility of controlling their data and relationships in a global way, but, in the described situation, an interoperability problem exists. This problem is called in the literature the *Walled Garden Problem* [60].

The approach suggested in this work is to establish data usage policy management and data storage in different domains. Administrators specify policies associated with data in a specific access control management service while data is stored in WBSNs. Therefore, the PDP is located in the first and the PEP is located in the second. The decoupling between both services should be so that access control management services do not know data details and data storage services do not know policy details; however, both type of services must count on a common data identification mechanism (note that it does not have to be for each independent data item, data sets are also appropriate) that allows the identification of the access control policies at stake given a certain access request. The decoupling should be also enough to allow a seamless application to use the same policies to different data sets independently of the WBSN in which data is stored in.

Access control tokens are a natural way to implement this approach. Particularly, the ITU-T X.812[10] recommendation describes tokens as elements possibly created by requesters and composed of multiple information. Besides, it is highlighted that they differ from certificates in the fact that certificates are delivered by a trusted authority. However, without losing generality and simplifying this issue, in this work, it is considered that tokens comprise tickets, certificates or any other elements which may provide the sufficient access control information to the PEP. Therefore, requesters are granted access only after they have acquired the appropriate token. Specifically, a requester first obtains the access token at the PDP and then, he presents this token at the PEP (which corresponds to the WBSN) to access to the requested data. If the PEP can extract directly from the tokens the usage decision taken by the PDP, no more interactions are needed. However, depending of the specific design of tokens, an interaction between the PEP and the PDP may be needed. The architecture described here is depicted in Figure 4(b).

As in the previous basic mechanism (Section 5.2), features involved in policy construction are user, data and relationships attributes, and actions. However, there is a fundamental difference between the previous mechanism and this one, tokens can have a specific duration attached to them and once a token is acquired it can be used in several data requests until it expires. In this case the duration can consider an absolute time period or relative to the token creation.

[10] http://www.itu.int/rec/T-REC-X.812/en, last access November 2012

Interestingly, in this mechanism it must be pointed out the way in which tokens are delivered. They can be delivered through out of band mechanisms such as emails, phone calls and so on, or through a specific entity specially developed to achieve this goal. Besides, bearing in mind that tokens are usually valid for a particular period of time, the main way to distribute them is through an entity, for example called token management service.

In regard to administrative operations, administrators have to establish policies in the token service and relate them to the data stored in the chosen storage service. Indeed, this issue can be performed through multiple ways, for example using a table in the token service to associate policies, tokens and data. Afterwards, when revocation is required different strategies can be applied. Tokens can be re-distributed or, due to the expiration time attached to them, requesters may, when necessary, acquire the corresponding new one.

This mechanism provides substantial benefits regarding the rest of mechanisms. It provides flexibility because policies are separated from data and they can be managed independently. Moreover, this decentralization leads to the reduction of overheads in the storage device. Tokens are valid during a period of time and policies do not need to be checked in each access request. Furthermore, this mechanism provides a simple technique to control revocation due to time stamps attached to each token. Nonetheless, there are some disadvantages. First, as in the basic approach, the problem of honest-but-curious storage services persists. Second, an entity to manage tokens is required. Thus, it is assumed an entirely trusted storage device and a trusted entity that acts as a token provider. In the end, more interactions are involved in this access control mechanism, making the process a little harder to perform. In particular, communications between the user, the token service and the storage device are expected.

A chief example of recent efforts to design this type of mechanisms is the one developed by the User-Managed Access (UMA) Working Group which presents an architecture, called UMA[11], to give users control over their personal data. This architecture focuses on keeping data separated from its access control management and requires a token to access data. Besides, another challenging example, directly related to WBSNs, is PrPl [41]. In this approach users store data in their personal computers or in chosen servers, called *butlers*, and the access is managed through the delivery of tickets, which are elements equivalent to tokens.

5.4 Addressing sticky policies

According to the *sticky policies* requirement, users must be able to control the usage of their data after access to it has been granted. A feasible way of facing this requirement is through the continuous performance of policy evaluation and enforcement, avoiding conflicts such as the execution of not

[11] http://kantarainitiative.org/, last access November 2012

granted activities (e.g., the printing of an object in which just a read permission is attached) [23]. However, this simple technique may produce high load of traffic or overload in WBSN servers, being indispensable the development of more efficient procedures. The fulfilment of this requirement is directly related to the specific architecture selected for implementing the access control mechanism.

Generally, to guarantee a full protection of data, some type of reference monitor must be deployed in the client (i.e., the location where data is going to be used). If combined with the basic approach described in Section 5.2, an architecture such as the one depicted in Figure 4(c) can be used. The reference monitor in the client will be in charge of acting accordingly to the indications of the reference monitor in the server and then, if a change in the policies (revocation rights) affects the data in use, the client reference monitor has to carry out the appropriate verifications by connecting to the server. In other words, if revocation of a requester (v_i) is performed in respect to a certain data (d_j) when v_i is using d_j, d_j must be immediately unavailable to v_i.

Indeed, any WBSN mechanism can (partially) fulfil this requirement but taking the following matter into account. As social networks are composed of web elements, an inherent property is that the web site content, once visualized, is available until reloading the page. Due to that fact, if v_i is using d_j and simultaneously access permission to d_j is denied to v_i, v_i will be able to manage d_j until the page is reloaded. A full compliance with the *sticky policy* paradigm would mean that after the change is made on the policies, their effects are immediately enforced.

5.5 Addressing data exposure minimization

Servers store information using its own techniques and users are not directly involved in controlling data which may lead to unknown disclosures of information. As a result, the main applicable countermeasure is cryptography, providing therefore a cryptographic access control mechanism (called 'lock and key' in Section 2).

Ideally, the storage service (the WBSN) has no access to clear data or policies associated to it. In fact, as mentioned in Section 2, under this approach a reference monitor is not specifically needed because data administrators encrypt data and it is stored encrypted in the storage service. Requesters may freely access to encrypted data, as only having the appropriate decryption key would grant them access to data in clear.

One of the approaches that can be taken is that administrators create user keys according to the keys used to encrypt the data whose access needs to be granted. The other approach, more complex, is the one considered by Attribute Based Encryption (ABE)d or any cryptographic encryption algorithm in which keys or ciphertexts involve a wide range of attributes. Anyhow, in both cases a user key is delivered to the user. Then, analogously to token based access control (see Section 5.3), the way of delivering keys can take different forms. As in the previous case, they can be delivered through out of band mechanisms or through a specific entity. Without losing generality, it is assumed that an entity called key management service distributes available keys, and such architecture is depicted in Figure 4(d).

Besides, in respect to revocation, it can be managed on different ways. In general, administrators have to create new keys, perform the appropriate re-encryptions and distribute the new ones. However, depending on the approach this scheme can be modified, for example, a proxy can be used to perform the appropriate re-encryptions [61].

Lastly, pros and cons of this mechanism are pointed out. On the one hand, this mechanism has a relevant benefit that is users control their own data. On the contrary, there are several drawbacks. Revocation can be quite tedious due to re-distributions or re-encryptions. Also, as in token based access control, it is indispensable trusting the entity which manages keys. Eventually, due to the use of cryptography, performance is reduced, though depending on the approach and even causing minimal negative effects.

Notice that apart from cryptography which is the most straightforward technique to devise, other ones could be applied (e.g. steganography), though they have not received significant research attention yet.

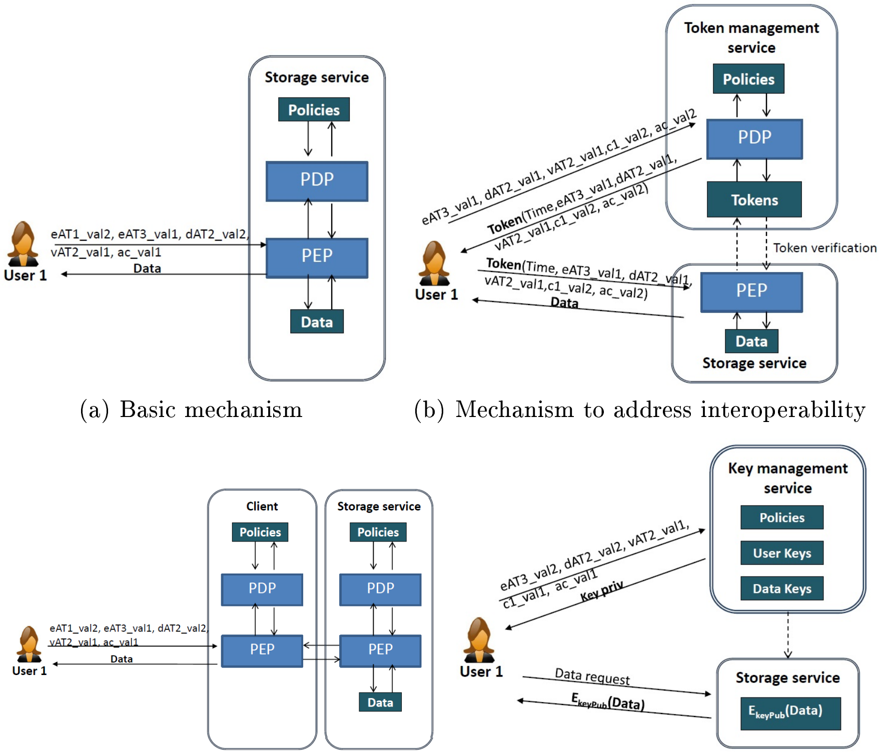
6 Analysis of web based social networks

Since the emergence of WBSNs, awareness of the importance of privacy has been steadily increasing. One of the goals of this work, which refers to the last contribution, is to analyze recent advances in user-managed access control systems for WBSN. Therefore, in this Section, a set of 25 academic proposals that work on privacy regarding access control in WBSNs and 9 of the most currently used and active WBSNs are analyzed in regard to their adaptation to the identified WBSN's requirements, i.e., relationship-based, fine-grained, interoperability, sticky policies and data exposure minimization.

Tables 1 and 2 summarize the results of the analysis. In general terms, academic proposals largely differ from WBSNs in use. Academic contributions are specially focused on cryptographic access control enforcement and the development of novel techniques to provide user-managed data control, although leaving aside sticky policies. By contrast, the majority of active WBSNs tend to manage access control in a simpler way, neglecting important requirements such as *interoperability* or *data exposure minimization*. Moreover, it has to be noticed that neither academic proposals nor WBSNs in use, except for Badoo, handle conditions or obligations.

In order to structure this analysis, in each proposal and WBSN in use, the following ten features have been examined:

- **Elements in authorization predicates**: as mentioned in Section 5, authorization predicates should be composed of rights and user, data and



(a) Basic mechanism

(b) Mechanism to address interoperability

(c) Mechanism to address sticky policies

(d) Mechanism to address data minimization exposure

Fig. 4. Proposed access control mechanisms

relationship attributes, which is directly related to *relationship-based* and *fine-grained* requirements. In particular, the considered attributes of (direct) relationships are direction, roles, level of trust, duration and own features such as history or creation time. Regarding indirect relationships, as mentioned in Section 3, attributes are generally derived from those of the direct ones. This issue is noted with the symbol γ in Table 1. Mainly, to manage indirect relationships, academic proposals use a particular feature called *depth* [62, 63, 64, 65] that is included in policies (this is the feature referred to as *length* previously). Similarly, Facebook, Picasa, MySpace and Flickr, also manage this issue but without giving details of the internal procedure. Data attributes considered in the analysis are data privacy (user assigned label) and own features like creation time. Moreover, considered user attributes are location and profile data. In some approaches, an identification attribute may be considered for data, users and relationships. Although, this attribute is not explicitly used in academic proposals. Likewise, it is also necessary to highlight that rights are only pointed out if they are considered in the policy itself. Indeed, rights are always involved in access control. Once giving access to a particular object, at least the read permission is granted. However, in these cases, in which rights are not directly involved in policies, the management of rights is not distinguished.

- **Interoperability**: indication of the management of tokens which, as previously studied, is a key step towards interoperability.
- **Sticky policies**: indication of the fact that policies follow the data to which they apply.
- **Data exposure minimization**: indication of a technique to prevent storage services from accessing data that users have not authorized. As pointed out in Section 5.5, this issue can be solved by different techniques but it is cryptography the only one applied. Therefore, the analysis of this feature refers to the identification of having or not used cryptography.
- **Policy definition**: specification of languages or tools used to develop policies, like XML.
- **Token element (if required)**: element used as a token. For instance, certificates, tickets and so on.
- **Key element (if required)**: element(s) used as keys, specifying required types. For instance, a pair of public/private keys.
- **Details on policy evaluation and enforcement mechanisms**: description of ways to carry out policy evaluation and specific enforcement mechanisms.
- **Revocation techniques**: revocation considers the modification of policies. This issue can be commonly managed by any technique that makes use of policies by simply changing them. Nonetheless, in this analysis, revocation is exclusively studied in academic proposals and WBSNs in use that explicitly mention having tackled this issue.

  In general terms, techniques to perform revocation are classified as the following groups:
  - Changing policies: revocation can be simply carried out by changing policies attached to data, users or relationships.
  - Expiration time: revocation can be performed by associating an expiration time to access control tokens. Consequently, once it expires, a new token is required.
  - Key updates: once a change in the policy is produced, keys are updated and appropriately distributed.
  - Re-encryption: using cryptography may imply several encryption and re-encryption operations. Then, once a change in the policy is produced, re-encryptions are indispensable.
  - Through an external element: revocation can be managed outside the system by making use of external elements. For example, it can be carried out through a revocation list, a proxy, a PKI, etc.
  - Through a particular scheme: revocation can be carried out through specific schemes such as particular protocols or cryptographic algorithms.

- **Trusted entity**: it can be a user, a group of users or a particular device, for instance, a storage device.

Furthermore, it is noticeable that all academic proposals and WBSNs in use manage the *relationship-based* requirement. However, even considering the management of relationships attributes, the establishment of relationships is generally managed out of band. For instance, the administrator directly communicates keys/tokens to his contacts regarding the level of trust put in them. A more practical example can be identified in the WBSN Picassa, where administrators send to his contacts the URL of his photo albums.

Finally, there are a pair of symbols to notice. Specifically, symbol "X" refers to the fact that a given approach has explicitly mentioned that this aspect is out of the scope of the proposal, and symbol "-" implies that such approach does not point out anything about a specific issue.

6.1 Academic proposals for WBSNs

A total of 25 academic proposals focus on privacy in WBSNs. The summary of the analysis is presented in Table 1. In general terms they can be divided into 3 groups, namely, (1) the approaches exclusively associated with the basic mechanism for implementing $SoNeUCON_{ABC}$, (2) contributions which may satisfy the *interoperability* requirement, and (3) proposals particularly focused on cryptographic access control enforcement and which are linked to the *data exposure minimization* requirement.

Three of the studied proposals fall into the first group and they have been proposed from 2006 to 2009 [62, 63, 66]. All of them follow a server-side architecture. In this case, the most remarkable feature is the fact that Access Control Lists (ACLs) are used in 2 out of 3 approaches in order to manage policy decision and enforcement. ACLs seem to be a common and feasible way to manage access control in this basic approach. Besides, it is surprising that not a single proposal describes some kind of revocation technique.

In respect to *interoperability*, 9 contributions meet this requirement and the architecture is also server-side [67, 41, 68, 40, 69, 70, 43, 71, 72]. In general, requesters obtain a token which can be represented in multiple ways, such as XML files [68] or tickets [41], and can be composed of assorted features, like keys [68] or digital signatures [40]. However, some of these contributions [40, 69, 70, 43] also manage policies in the storage service, which may prevent them from being interoperable. This is equivalent to the basic mechanism and, even using tokens, these approaches have to be further studied to identify the possibility of adapting them to satisfy the *interoperability* requirement. One relevant common point in some proposals is the use of tokens to attest relationships and their attributes [69, 40, 67, 68, 72]. It matches perfectly with the fact that relationships chief elements of WBSNs. Indeed, despite requiring future work, relationship certificates can be a promising mechanism to manage relationships in the way that relationships remain hidden from all users but to the pair of them who are involved in it [46]. Lastly, notice that revocation is managed by the majority of approaches and techniques applied are quite assorted. For instance in PrPl tokens have attached an expiration time and once expired new tokens have to be created [41].

Only three proposals address the *sticky policy* requirement. Specifically, all of them make use of ABE and, as mentioned in Section 5.4, ABE cryptography relies on the involvement of policies in keys or ciphertexts, which naturally helps to control access to data wherever it is used [64, 61, 73].

A total of 15 recent proposals make use of cryptography before storing data in the server, therefore, they are in the position of fulfilling the *data exposure minimization* requirement [74, 75, 76, 42, 77, 78, 39, 79, 64, 61, 73, 80, 71, 72, 81]. The overall procedure is based on the use of a public-private key pairs in multiple and assorted algorithms according to the defined policy decision and enforcement functions. However, [81] proposes an hybrid scheme in which data is symmetrically encrypted and asymmetric cryptography is applied in the data acquisition and decryption process. Some proposals focus on requiring a particular policy in the key creation process [64, 61, 73], requiring proxies to forward data among different user groups [61] or requiring methods to store fake information [74, 77, 80]. Other identified issue is that the most common way of managing revocation is modifying keys and performing the appropriate re-encryptions.

A noticeable feature common to all proposals is the management of unidirectional relationships without considering a direction attribute as an authorization predicate element. Furthermore, this matter emphasises that access control mechanisms are established from one user to another and the particular implementations are the ones which make relationships bidirectional.

One last point for this analysis is that approaches that do not handle cryptography require to trust multiple entities. However, there are some exceptions and several cryptographic proposals also trust particular devices but it is not due to lack of data confidentiality but to preserve data from unauthorized deletions [42], from dishonest revocations [61, 73] or from unfairly key managements [79].

6.2 WBSNs in use

Many WBSNs are currently used by thousands of people. Nevertheless, for the sake of simplicity, 9 of the most representative WBSNs in use have been studied according to features described above. The overall results are summarized in Table 2.

Firstly, it can be noticed that basic architectures similar to the one proposed in Section 5.2 are the most used, although the implemented access control models are much simpler than $SoNeUCON_{ABC}$. Moreover, it is remarkable the existence of three WBSNs that manage tokens and are candidates for satisfying interoperability, namely, Picasa, MySpace and Flickr. Picasa, focused on photos management, is exclusively based on tokens which take the form of URLs. Likewise, Flickr, that also focused on photo management, uses URLs to provide access to photos. This WBSN is specially relevant concerning the management of public photos because everybody is able to access through the appropriate URL. Moreover, Flickr allows the management of rights in regard to notes, commentaries and photos. Similarly, MySpace, a WBSN to share data such as photos, videos or music, uses URLs as tokens as well. However, MySpace only uses tokens for photos, applying simpler techniques for other types of data like wall messages. In sum, taking into consideration this analysis, tokens seems to be specially valuable regarding photo management.

In relation to attributes of authorization predicates that are directly related to the *fine-grained* requirement, there is a common pattern in which the role attribute (of relationships) and the data privacy attribute (of data elements) are key points. Besides, all analyzed WBSNs manage direction attribute in a bidirectional way, except for Twitter in which administrators request users to be followers of them but not in the other way round. Interestingly, this issue highlights again that mechanisms are intrinsically unidirectional and the bidirectional nature is provided by implementations. By contrast, other attributes such as the creation time of a relationship, size of data or nationality of users, are left aside. Also, similar to academic approaches, WBSNs in use base on relationships in which the administrator is who directly communicates, out of band, keys or tokens.

Concerning revocation, the most of proposals manage it through revocation lists consisting of blocking revoked users from accessing to data. This is the simplest technique among the identified ones but the most used.

Finally, regarding trusted entities, it must be noted that WBSNs do not detail whether some entity or object must be trusted or not. Nonetheless, as there is no evidence of the fact that WBSNs address *data exposure minimization*, trust must be specially put on storage services (i.e., the WBSN), as well as, on users to whom tokens are delivered.

7 Conclusions and open research issues

One of the main problems that is currently attracting more attention in Web Based Social Networks (WBSNs) is the design and implementation of user-managed access control systems, as stated in [8]. For this purpose, four requirements have been previously identified. Moreover, in this work a fifth is added. This work contributes in this direction with several results. First, $SoNeUCON_{ABC}$, an access control model for WBSNs is proposed. This model allows to directly fulfil two of the requirements identified in [8], *relationship-based* and *fine-grained*. Based on this model, a set of mechanisms are selected so the three remaining requirements, *sticky policy*, *interoperability* and *data exposure minimization*, can be fulfilled. As a result of this discussion, a pair of relevant conclusions are reached. On the one hand, it is quite feasible to implement in the short term access control mechanisms for WBSNs that satisfy the requirements *relationship-based*, *fine-grained*, *interoperability* and *sticky policy*. On the other hand, it is also feasible, though computationally challenging, to implement in the medium term access control mechanisms for WBSNs that satisfy the requirements *relationship-based*, *fine-grained*, *sticky policy* and *data exposure minimization* (some of them with limitations).

More to the point, a total of 25 academic approaches and 9 WBSNs in use have been analyzed against the mentioned requirements. Summing up, academic approaches leave aside *fine-grained* and *sticky policy* requirements while weaknesses of WBSNs in use are *interoperability* and *data exposure minimization*. Additionally, the analysis shows that conditions and obligations are not generally addressed.

Nonetheless, a set of open issues are identified. Firstly, in this paper a general overview and applicability of $SoNeUCON_{ABC}$ is presented but its formalization is out of the scope of this work. For instance, regarding policy language and the corresponding policy construction, extended work is required. A first step towards an appropriate formalization is to follow approaches such as [23] or [54], related to $UCON_{ABC}$. Moreover, the formalization has to be supported by other authors specially focused on policy description [17]. Furthermore, indirect relationships must be particularly detailed by describing ways of calculating values of attributes involved [46]. For example, if the path between two nodes is n, the main point is to establish a method to enforce access control regarding the full relationship (path) that involves the n nodes. Therefore, who is the person that must establish these patterns, which factors must be considered or how particularly the calculus must be performed are some considerations to be determined.

Additionally, related to $SoNeUCON_{ABC}$ and the *fine-grained* requirement, WBSNs are systems which involve a huge quantity of users managing large sets of data by the establishment of access control policies. Then, a challenging issue is to identify demanding WBSN features, such as the possible specification of indirect relationships or cliques (eg. a close group of users), and look for expressive access control policies to achieve a successful fine-grained access control management. Due to that fact, the expressiveness of access control models regarding policy languages is a future step [82].

As recently identified by J. Park *et al.* [58, 57] the distinction between users and sessions is an appealing matter. Policies may be defined in regard to the user session. For instance, a user opens different sessions from different computers, which means from different IPs, and access control mechanisms provide him with different permissions in respect to the session. Thus, future work runs towards the study of novel approaches associated with this matter [58, 57] and their integration of this issue into our model.

Other relevant point in WBSNs is the difficult definition of users and their administration rights over data. This is related to co-ownership. Although for simplicity this paper recognizes as administrators any user who is the owner of an object or carries out administrative operations on it, differences between both must be identified. The following step then will be to define techniques to face co-ownership problem [83].

Furthermore, an interesting point refers to privacy according to the discovery of the WBSN structure. In other words, the identification of users and their relationships is a challenging problem. Some contributions and the whole of WBSNs in use do not consider the fact that relationships can be inferred from the social network structure.

Lastly, the main important issue to attain in future work is the satisfaction of the whole set of requirements. Indeed, bearing the persistent search of flexibility in mind and alluding to the *Walled Garden* problem, mechanisms based on tokens seem to be promising. Consequently, studies must be headed towards the analysis of the complete integration of tokens in the basic approach, as well as, the inclusion of cryptography to satisfy *data exposure minimization* requirement. As a final step, all developments must be appropriately designed and evaluated.

References

1. Boyd, D.M., Ellison, N.B.: Social network sites: Definition, history, and scholarship. Journal of Computer−Mediated Communication **13** (2007) 210–230
2. Parent, W.A.: Privacy, morality, and the law. Philosophy and Public Affairs **12** (1983) 269–288
3. Becker, J., Chen, H.: Measuring Privacy Risk in Online Social Networks. In: Proc. of W2SP 2009: Web 2.0 Security and Privacy. (2009)
4. Acquisti, A., Gross, R.: Imagined Communities: Awareness, Information Sharing, and Privacy on the Facebook. In: Privacy Enhancing Technologies. Volume 4258 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2006) 36–58
5. Oracle-Team: Online security, A Human Perspective. (2011)
6. Dey, R., Jelveh, Z., Ross, K.W.: Facebook users have become much more private: A large-scale study. In: Proc. of SESOC 2012. (2012)
7. Dwyer, C., Hiltz, S.R.: Trust and Privacy Concern Within Social Networking Sites : A Comparison of Facebook and MySpace. Information Systems (2007)
8. Carrie, D., Gates, E.: Access control requirements for web 2.0 security and privacy. In: Proc. of Wks. on Web 2.0 Security & Privacy (W2SP 2007. (2007)
9. Gao, H., Hu, J., Huang, T., Wang, J., Chen, Y.: Security issues in online social networks. IEEE Internet Computing **15** (2011) 56–63
10. Ajami, R., Ramadan, N., Mohamed, N., Al-Jaroodi, J.: Security challenges and approaches in online social networks: A survey. Intl. Journal of Computer Science and Network Security **11** (2011) 1–12
11. Zheleva, E., Getoor, L.: Privacy in Social Networks: A Survey. In: Social Network Data Analytics. Springer (2011)
12. Carminati, B., Ferrari, E.: Access control and privacy in web-based social networks. International Journal of Web Information SystemsVol4 **4** (2008) 395–415
13. Sastry, M., Krishnan, R., Sandhu, R.: A New Modeling Paradigm for Dynamic Authorization in Multi−domain Systems. (2007) 153–158

14. Sandhu, R.S., Samarati, P.: Access control: Principles and practice. Access (1994) 40–48
15. Razavi, M.N., Iverson, L.: Towards usable privacy for social software. Technical report, University of British Columbia (2007)
16. Bertino, E., Bonatti, P.A., Ferrari, E.: Trbac: a temporal role−based access control model. In: Symposium on Access Control Models and Technologies. Proc. of the fifth ACM Wks. on Role−based access control, ACM (2000) 21–30
17. Fong, P.W.L.: Relationship−based access control: protection model and policy language. In: Proc. of the first ACM conference on Data and application security and privacy. CODASPY '11, ACM (2011) 191–202
18. Giunchiglia, F., Zhang, R., Crispo, B.: Relbac: Relation based access control. In: Semantics, Knowledge and Grid, 2008. SKG '08. Fourth International Conference on. (2008) 3 –11
19. Ray, I., Kumar, M., Yu, L.: LRBAC: A Location−Aware Role−Based Access Control Model. In: Information Systems Security. Volume 4332 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2006) 147–161
20. Covington, M., Moyer, M., Ahamad, M.: Generalized role−based access control for securing future applications. In: 23rd National Information Systems Security Conference, Citeseer (2000)
21. Capitani di Vimercati, S., Foresti, S., Samarati, P.: Authorization and Access Control. Security, Privacy, and Trust in Modern Data Management (2007) 39–53
22. Scholl, M., Steinberg, D.: Security Architecture Design Process for Health Information Exchanges (HIEs). (Solutions)
23. Lazouski, A., Martinelli, F., Mori, P.: Usage control in computer security: A survey. Computer Science Review **4** (2010) 81–99
24. Salim, F., Reid, J., Dawson, E.: An administrative model for UCONabc. In: Proc. of the Eighth Australasian Conference on Information Security. Volume 105 of AISC '10. (2010) 32–38
25. Bishop, M.: Computer Security Art and Science. Addison-Wesley (2003)
26. Mun, M., Hao, S., Mishra, N., Shilton, K., Burke, J., Estrin, D., Hansen, M., Govindan, R.: Personal data vaults: a locus of control for personal data streams. In: Proc. of the 6th International COnference. Co−NEXT '10, ACM (2010) 17:1–17:12
27. Shilton, K., Burke, J.A., Estrin, D., Hansen, M.: Designing the Personal Data Stream : Enabling Participatory Privacy in Mobile Personal Sensing. Work (2009) 25–27
28. Bouganim, L., Pucheral, P.: Chip−secured data access: confidential data on untrusted servers. In: Proc. of the 28th international conference on Very Large Data Bases. VLDB '02, VLDB Endowment (2002) 131–142
29. Allard, T., Anciaux, N., Bouganim, L., Guo, Y., Le Folgoc, L., Nguyen, B., Pucheral, P., Ray, I., Ray, I., Yin, S.: Secure personal data servers: a vision paper. **3** (2010) 25–35
30. di Vimercati, S.D.C., Foresti, S., Jajodia, S., Paraboschi, S., Samarati, P.: A data outsourcing architecture combining cryptography and access control. In: Proc. of the 2007 ACM Wks. on Computer security architecture. CSAW '07, ACM (2007) 63–69
31. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext−policy attribute−based encryption. In: Proc. of the 2007 IEEE Symposium on Security and Privacy. SP '07, IEEE Computer Society (2007)

32. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute−based encryption for fine−grained access control of encrypted data. In: Proc. of the 13th ACM conference on Computer and communications security. CCS '06, ACM (2006) 89–98
33. Chase, M.: Multi−authority attribute based encryption. In: Proc. of the 4th conference on Theory of cryptography. TCC'07, Springer−Verlag (2007) 515–534
34. Attrapadung, N., Imai, H.: Conjunctive broadcast and attribute−based encryption. In: Proc. of the 3rd International Conference Palo Alto on Pairing−Based Cryptography. Pairing '09, Springer−Verlag (2009)
35. Chase, M., Chow, S.S.: Improving privacy and security in multi−authority attribute−based encryption. In: Proc. of the 16th ACM conference on Computer and communications security. CCS '09, ACM (2009) 121–130
36. Lin, H., Cao, Z., Liang, X., Shao, J.: Secure threshold multi authority attribute based encryption without a central authority. Inf. Sci. **180** (2010) 2618–2632
37. Shi, W.: Attribute Based Encryption with Pattern−awareness by Attribute Based Encryption with Pattern-awareness. Master's thesis, Inha University (2010)
38. Ostrovsky, R., Sahai, A., Waters, B.: Attribute−based encryption with non-monotonic access structures. In: Proc. of the 14th ACM conference on Computer and communications security. CCS '07, ACM (2007) 195–203
39. Zhu, Y., Hu, Z., Wang, H., Hu, H., Ahn, G.j.: A Collaborative Framework for Privacy Protection in Online Social Networks. Organization (2010) 1–15
40. Tootoonchian, A., Saroiu, S., Wolman, A.: Lockr : Better Privacy for Social Networks. Design (2009)
41. Seong, S.W., Seo, J., Nasielski, M., Sengupta, D., Hangal, S., Teh, S.K., Chu, R., Dodson, B., Lam, M.S.: Prpl: a decentralized social networking infrastructure. (2010) 8:1–8:8
42. Buchegger, S., Schiöberg, D., Vu, L.H., Datta, A.: Peerson: P2p social networking: early experiences and insights. (2009) 46–52
43. Backes, M., Maffei, M.: A security API for distributed social networks. Network and Distributed System Security (2011)
44. Harary, F., Norman, R.Z.: Graph theory as a mathematical model in social science. (1953)
45. Carminati, B., Ferrari, E.: Access control and privacy in web−based social networks. International Journal of Web Information Systems **4** (2008)
46. Carminati, B., Ferrari, E., Perego, A.: Private relationships in social networks. In: Proc. of the 2007 IEEE 23rd International Conference on Data Engineering Wks., IEEE Computer Society (2007) 163–171
47. Carminati, B., Ferrari, E., Perego, A.: Rule−Based Access Control for Social Networks. In: Proc. OTM 2006 Workshops (On the Move to Meaningful Internet Systems). Volume 4278 of LNCS., Springer (2006) 1734–1744
48. Carminati, B., Ferrari, E.: Privacy−aware collaborative access control in web−based social networks. In: Proceeedings of the 22nd annual IFIP WG 11.3 working conference on Data and Applications Security, Springer−Verlag (2008) 81–96
49. Schneier, B.: A taxonomy of social networking data. Security Privacy, IEEE **8** (2010)
50. Squicciarini, A.C., Shehab, M., Paci, F.: Collective privacy management in social networks. In: Proc. of the 18th international conference on World wide web. WWW '09, ACM (2009) 521–530

51. Squicciarini, A.C., Shehab, M., Wede, J.: Privacy policies for shared content in social network sites. The VLDB Journal (2010) 777–796
52. Covington, M.J., Sastry, M.R.: A contextual attribute-based access control model. In: Proc. of the 2006 international conference on On the Move to Meaningful Internet Systems: AWeSOMe, CAMS, COMINF, IS, KSinBIT, MIOS-CIAO, MONET - Volume Part II. OTM'06 (2006) 1996–2006
53. Nin, J., Carminati, B., Ferrari, E., Torra, V.: Computing reputation for collaborative private networks, IEEE Computer Society (2009) 246–253
54. Park, J., Sandhu, R.: The UCONabc usage control model. ACM Trans. Inf. Syst. Secur. **7** (2004) 128–174
55. Yuan, E., Tong, J.: Attributed Based Access Control (ABAC) for Web Services. In: Proc. of the IEEE International Conference on Web Services. ICWS '05, IEEE Computer Society (2005) 561–569
56. Shen, H., Hong, F.: An Attribute-Based Access Control Model for Web Services. In: Parallel and Distributed Computing, Applications and Technologies, 2006. PDCAT '06. Seventh International Conference on. (2006) 74 –79
57. Park, J., Sandhu, R.: A Position Paper: A Usage Control (UCON) Model for Social Networks Privacy. (2000)
58. Park, J., Sandhu, R., Cheng, Y.: A user−activity−centric framework for access control in online social networks. Internet Computing, IEEE **15** (2011) 62 –65
59. Zhang, X., Park, J., Parisi-Presicce, F., Sandhu, R.: A logical specification for usage control. In: Proc. of the ninth ACM symposium on Access control models and technologies. SACMAT '04, ACM (2004) 1–10
60. man Au Yeung, C., Liccardi, I., Lu, K., Seneviratne, O., Berners−Lee, T.: Decentralization: The future of online social networking. In: W3C Wks. on the Future of Social Networking Position Papers. (2009)
61. Jahid, S., Mittal, P., Borisov, N.: Easier: encryption−based access control in social networks with efficient revocation. In: Proc. of the 6th ACM Symposium on Information, Computer and Communications Security. ASIACCS '11, ACM (2011) 411–415
62. Kruk, S., Grzonkowski, S., Gzella, A., Woroniecki, T., Choi, H.C.: D−foaf: Distributed identity management with access rights delegation. In: The Semantic Web ? ASWC 2006. Volume 4185 of Lecture Notes in Computer Science., Springer Berlin / Heidelberg (2006) 140–154
63. Carminati, B., Ferrari, E., Heatherly, R., Kantarcioglu, M., Thuraisingham, B.: A semantic web based framework for social network access control. In: Proc. of the 14th ACM symposium on Access control models and technologies. SACMAT '09, ACM (2009) 177–186
64. Baden, R., Bender, A., Spring, N., Bhattacharjee, B., Starin, D.: Persona: an online social network with user−defined privacy. SIGCOMM Comput. Commun. Rev. **39** (2009) 135–146
65. Kourtellis, N., Finnis, J., Anderson, P., Blackburn, J., Borcea, C., Iamnitchi, A.: Prometheus : User−Controlled P2P Social Data Management for Socially−Aware Applications. Ifip International Federation For Information Processing (2010) 212–231
66. Besmer, A., Lipford, H.R., Shehab, M., Cheek, G.: Social applications: exploring a more secure framework. In: Proc. of the 5th Symposium on Usable Privacy and Security. SOUPS '09, ACM (2009) 2:1–2:10
67. Ali, B., Villegas, W., Maheswaran, M.: A trust based approach for protecting user data in social networks. (2007) 288–293

68. Conti, M., Hasani, A., Crispo, B.: Virtual private social networks. In: Proc. of the first ACM conference on Data and application security and privacy. CODASPY '11, ACM (2011) 39–50
69. Aiello, L.M., Ruffo, G.: Secure and flexible framework for decentralized social network services. 2010 8th IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops) (2010) 594–599
70. Shakimov, A., Lim, H., Li, K., Liu, D., Varshavsky, A.: Vis−a−Vis : Privacy−Preserving Online Social Networking via Virtual Individual Servers. (2010)
71. Ackermann, M., Ludwig, B., Hymon, K., Wilhelm, K.: Helloworld: An open source, distributed and secure social network. In: W3C Wks. on the Future of Social Networking. (2009)
72. Aiello, L.M., Ruffo, G.: Lotusnet: Tunable privacy for distributed online social network services. Comput. Commun. **35** (2012) 75–88
73. Jahid, S., Nilizadeh, S., Mittal, P., Borisov, N., Kapadia, A.: Decent: A decentralized architecture for enforcing privacy in online social networks. (2012)
74. Guha, S., Tang, K., Francis, P.: Noyb: privacy in online social networks. In: Proc. of the first Wks. on Online social networks. WOSN '08, ACM (2008) 49–54
75. Lucas, M.M., Borisov, N.: Flybynight: mitigating the privacy risks of social networking. In: Proc. of the 7th ACM Wks. on Privacy in the electronic society. WPES '08, ACM (2008) 1–8
76. Cutillo, L.A., Molva, R., Strufe, T.: Safebook: Feasibility of transitive cooperation for privacy on a decentralized social network. 2009 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks & Workshops (2009) 1–6
77. Luo, W., Xie, Q., Hengartner, U.: FaceCloak: An Architecture for User Privacy on Social Networking Sites. 2009 International Conference on Computational Science and Engineering (2009) 26–33
78. Anderson, J., Diaz, C., Stajano, F., Bonneau, J.: Privacy−Enabling Social Networking Over Untrusted Networks Categories and Subject Descriptors. (Security) 2–7
79. Frikken, K.B., Srinivas, P.: Key−allocation schemes for private social networks. In: Proc. of the 8th ACM Wks. on Privacy in the electronic society. WPES '09, ACM (2009) 11–20
80. Besenyei, T., Földes, A., Gulyás, G., Imre, S.: StegoWeb: Towards the Ideal Private Web Content Publishing Tool. In: SECURWARE 2011, The Fifth International Conference on Emerging Security Information, Systems and Technologies. (2011) 109–114
81. Graffi, K., Groß, C., Stingl, D., Hartung, D., Kovacevic, A., Steinmetz, R.: Lifesocial.kom: A secure and p2p-based solution for online social networks. In: Proc. of the IEEE Consumer Communications and Networking Conference, IEEE Computer Society Press (2011)
82. Carreras, A., Rodriguez, L., Delgado, J., Maronas: Access control issues in social networks. (2010) 47–52
83. Squicciarini, A.C., Shehab, M., Paci, F.: Collective privacy management in social networks. In: Proc. of the 18th international conference on World wide web. WWW '09, ACM (2009) 521–530

Table 1. Analysis of academic proposals related to access control in WBSNs

						Features				
Proposals	**Elements in authorization predicates**	**Interoperability**	**Sticky Policies**	**Data exposure minimization**	**Policy definition**	**Token element (If required)**	**Keys (If required)**	**Details on policy evaluation and enforcement mechanisms**	**Revocation techniques**	**Trust entity**
D-FOAF (2006) [62]	eAT(trust, γ)	-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements used an ACL	-	Group of users
A semantic web based framework (2009) [63]	eAT(role, trust, γ) + Rights	-	-	-	SWRL	-	-	Resource granted if satisfying policies according to policy elements	-	-
Granular framework (2009) [66]	eAT(role) + vAT	-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	-	-
SAC (2007) [67]	eAT(trust)	√	-	-	-	Keys	Key portions	Resource is granted if having the appropriate token	-	Device (storage)
PrPl (2010) [41]	vAT + Rights	√	-	-	-	Tickets	Key pair(private/ public) per user	Resource is granted if having the appropriate token	Expiration time	Device (storage)
FaceVPSN (2011) [68]	-	√	-	-	-	XML file	-	Resource is granted if having the appropriate token	-	-
Lockr (2009) [40]	eAT(role)	√	-	-	-	Attestations	Key pair(private/ public) per user	Resource is granted if having the appropriate token and verifying the the token against special list	Re-sending tokens, through a external element (revocation list) or expiration time	-
Secure & Flexible framework (2010) [67]	eAT(role)	√	-	-	-	Certificate	Key pair(private/ public) per user	Resource is granted if having the appropriate token and verifying the regular expression attached to the token	Through a external element (certificate) attached a expiration time	-
Vis-a-Vis (2010) [70]	vAT	√	-	-	-	Group descriptor	Key pair(private/ public) per user	Resource is granted if having the appropriate token and satisfying policies according to policy elements	-	Group of users + Device(elements provide by third parties)
Securing Social Networks (2011) [43]	eAT(role) + Rights	√	-	-	Specific API	Signed pseudonym	Not implicitly specified	Resource is granted if having the appropriate token and verifying the token against a special list	Through a particular scheme (broadcast encryption)	X
NOYB (2008) [74]	-	-	-	√	-	-	Depending on web services	Cryptographic algorithm based on dictionaries	Key updates + Re-encrypting	Group of users + Device (elements provided by third parties)
FlyByNight (2008) [75]	-	-	-	√	-	-	Key pair(private/ public) per user, other group pair and proxy keys per user per group	Cryptographic algorithm with proxy cryptography to handle "one-to-many" requests	Key updates + Re-encrypting	-
Safebook (2009) [76]	eAT(trust)	-	-	√	-	-	Key pair(private/ public) per user, attribute	Cryptographic public key algorithm and certificates	Key updates + Re-encrypting	Group of users
PeerSoN (2009) [42]	eAT(role)	-	-	√	-	-	Not implicitly specified	Cryptographic algorithm	Through a external element (PKI)	-
FaceCloak (2009) [77]	eAT(role)	-	-	√	-	-	A master, an index and an access key per user and a master and index key per requester	Cryptographic algorithm	X	Group of users + Device (server)
Collaborative framework (2010) [39]	-	-	-	√	-	-	Key pair(private/ public) per user and other group pair	Cryptographic algorithm based on data blocks	Through a external element (revocation list)	Group of users
Key-allocation scheme (2009) [79]	-	-	-	√	-	-	Key pair(private/ public) per user and as many keys as the maximum path length	Cryptographic algorithm based on satisfying indirect paths	Key updates	Device (server)
LotusNet (2012) [72]	-	√	-	√		Certificate	Key per user	Certificate validation	Expiration time	Group of users
Persona (2009) [64]	eAT(role, γ) + vAT + Rights	-	√	√	Using attributes and rights	-	Key pair(private/ public) per user and a key per group	Cryptographic algorithm based on CP-ABE	Key updates+ Re-encrypting	Device (Key storage)
EASiER (2011) [61]	eAT(role)	-	√	√	Using attributes	-	Key pair(private/ public) per user and proxy keys depending on attributes and users	Cryptographic algorithm based on CP-ABE	Through a external element (proxy)	Proxy
DECENT (2012) [73]	eAT(role)	-	√	√	Using attributes	-	Key pair(private/ public) per user depending on attributes and users and a signature key	Cryptographic algorithm based on CP-ABE	Through a external element (proxy)	
StegoWeb (2011) [80]	-	-	-	√	-	-	-	Cryptographic algorithm	Key updates	-
Prometheus (2010) [65]	eAT(trust, role, γ)	-	-	√	-	-	Key pair(private/ public) per object or a key per object	Resource granted if satisfying policies according to policy elements and cryptographic public key algorithm	Through a external element (revocation list)	Group of users
LifeSocial.KOM (2011) [81]	-	-	-	√	-	-	Key pair(private/ public) per user and key per object	Cryptographic algorithm	-	-
Helloworld (2009) [71]	-	√	-	√*		Similar to OpenId URI	Key pair(private/ public) per user	Message interchange protocol	-	Group of users

* Data partially encrypted

Table 2. Analysis of access control features in WBSNs in use

					Features					
Proposals	**Elements in authorization predicates**	**Interoperability**	**Sticky Policies**	**Data exposure minimization**	**Policy definition definition**	**Token element (If required)**	**Keys (If required)**	**Details on policy evaluation and enforcement mechanisms**	**Revocation techniques**	**Trust entity**

LinkedIn (2003)[1]	eAT(role, γ)
	+ dAT(data privacy)
		-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	-	-

Hi5 (2003)[2]	eAT(role)
	+ dAT(data privacy)
	+ Rights
		-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	Through a external element (revocation list)	-

Facebook (2004)[3]	eAT(role) + Rights
	+ vAT(location)
	+ dAT(data privacy)
	+ Rights
		-	-	-	XACML	-	-	Resource granted if satisfying policies according to policy elements	Through a external element (revocation list)	-

Orkut (2004)[4]	eAT(role))
	+ vAT(email)
		-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	Through a external element (revocation list)	-

Badoo (2006)[5]	dAT(data privacy)	-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	Through a external element (revocation list)	-
Twitter (2006)[6]	dAT(data privacy)	-	-	-	-	-	-	Resource granted if satisfying policies according to policy elements	Through a external element (revocation list)	-

Picasa (2002)[7]	dAT(data privacy)
	+ eAT(γ)
		√	-	-	-	URL	-	Resource granted if satisfying the appropriate token	Through a external element (revocation list)	-

MySpace (2003)[8]	eAT(role, γ)
	+ vAT(age)
	+ dAT(data privacy)
		√	-	-	-	URL	-	Resource granted if satisfying policies according to policy elements or having the appropriate token	Through a external element (revocation list)	-

Flickr (2004)[9]	eAT(role, γ) + dAT(data privacy, own features) + Rights	√	-	-	XACML	URL	-	Resource granted if satisfying policies according to policy elements and/or having the appropriate token	Through a external element (revocation list)	-

[1] http://www.linkedin.com/ , last access November 2012
[2] http://www.hi5.com/ , last access November 2012
[3] http://www.facebook.com/ , last access November 2012
[4] http://www.orkut.com/PreSignup , last access November 2012
[5] http://badoo.com/ , last access November 2012
[6] https://twitter.com/ , last access November 2012
[7] https://picasaweb.google.com/home , last access November 2012
[8] http://www.myspace.com/ , last access November 2012
[9] http://www.flickr.com/ , last access November 2012