Privacy-enhancing Access Control Mechanism in Distributed Online Social Network

Robayet Nasim

Master's Thesis in Computer Science (30 ECTS credits)
at the Software Engineering of Distributed Systems Master's Program
Royal Institute of Technology year 2010
Supervisor at CSC was Sonja Buchegger
Examiner was Johan Håstad

Abstract

Dramatic growth in the number of subscribers in Online Social Networks (OSNs), such as Facebook, MySpace, Orkut, etc. shows their increasing popularity among people from different ages and sectors. However, currently, the users need to put complete trust on OSN service providers, to protect their sensitive information because of centralized access control at the providers. Taking advantage of this infrastructure, OSN service providers can expose their subscribers' personal information for targeted advertisements, or anything that is mentioned in the terms of the privacy agreement, including to change the terms. To give complete access control to the users over their data, there must be an alternative infrastructure, which removes dependence on OSN service providers. In order to address this privacy issue, Sonja Buchegger and Anwitaman Datta proposed 2-tier peer-to-peer architecture for social networks, called PeerSoN.

The goal of this master's thesis is to evaluate the suitability of eXtensible Access Control Markup Language (XACML) for Distributed Online Social Network (DOSN) access control and privacy preservation. To do that, firstly, we determine the requirements for access control in DOSN, and present a structure for users' profiles. Due to the wide ranges of requirements, we propose to use rule-based access control for the users in OSN, where the rules are based on both static and dynamic constraints. Secondly, in order to investigate whether these policies can be expressed in XACML or not, we implement some common authorization policies using SunXACML, an open source implementation of standard XACML version 2.0. Thirdly, to enhance privacy regarding authentication and enforcement, we offer to use secret key based authentication of SAML, and one of the XACML supported web or application servers, such as JBoss Application server, Fedora server, in conjunction with XACML. Finally, we evaluate our architecture against three types of attackers; namely, users from social links, users form outside of social links, and random persons, and claim that our mechanism is well protected against different threats, such as unauthorized access, impersonation attacks, identity theft, information leakage via friendship links, etc., specifically, when each user's profile is stored on his own machine.

Acknowledgements

I would like to express my utmost gratitude to my supervisor, Dr. Sonja Buchegger for providing me an opportunity to conduct my master's thesis under her, and also for her continuous guidance and support throughout the work. The discussions I have had with her are invaluable.

I would like to dedicate this thesis to my beloved parents, who were always with me with all their encouragements and prayers.

Contents

Chapter 1

Introduction

A social networking service is described as [49], “A Web site that provides a virtual community for people interested in a particular subject or just to "hang out" together. Members create their own online "profile" with biographical data, pictures, likes, dislikes, and any other information they choose to post. They communicate with each other by voice, chat, instant message, videoconference, and blogs, and the service typically provides a way for members to contact friends of other members”. Each member of any social network services communicates with each other through his/her profile, which generally contains information such as personal data, contact address, recent activities, pictures, etc., to represent a particular user. Exponential growth of the number of users in different online social networks (ONSs) indicates that it has become a popular platform for people to communicate and share information.

As online social networks (OSNs) deal with large amounts of personal information, they need appropriate security settings to protect this information form unauthorized access and unwanted disclosure. Social network providers already address this issue and offer privacy settings so that users can control access to their resources. Unfortunately, because of the centralized architecture (single authority) the users need to trust their providers blindly to enforce these access control settings. Further, there is almost no privacy offered against application providers in OSNs. As almost all OSNs are free for people to sign up, the question arises - how do these organizations support their storage infrastructure for their large number of subscribers? The answer is very easy to guess; by using their subscribers’ personal information such as demography, choices, etc. for advertisement. Also the centralized infrastructure of information repositories makes data mining from users’ profile much easier.

To the aim of addressing these security problems, Sonja Buchegger and Anwitaman Datta proposed a peer-to-peer (p2p) architecture for social networks, called **PeerSoN**[1]. This suggested architecture is based on two main considerations: “Privacy issues” and “Requirement of Internet connectivity for all transactions”. In PeerSoN, a distributed architecture removes the dependency on the service providers and also provides information exchange without Internet connection by taking advantage of opportunistic or delay-tolerant networking. However, the complete decentralization of the infrastructure raises research questions regarding privacy.

Authorization decides whether or not a person is eligible to perform an action on any resource. Authorization policies define constraints or requirements to get authorization permission. Online social networks usually define authorization policies, though not always, on the basis of users’ attributes such as trust, age, relationship, etc. Unfortunately, most of the policy languages are proprietary and also existing access control models are application-dependent. Moreover, in distributed environments, authorization components need to be interoperable, and also to col-

[1] www.peerson.net

laborate with each other, all peers need to agree on uniform authorization policies, such as syntax, semantics etc. Additionally, in OSN, because of diversity in both users and their access control requirements, expressiveness is one of the important parameters for design choice. In these types of dynamic environments, rule-based access control is more flexible than other access control models, where the rules are based on dynamic properties.

Although there are many languages that have been proposed so far for expressing access control policies XACML[2] (eXtensible Access Control Markup Language) gains most attention because of its standardization in the field of security, high expressiveness of constraints and also for providing a standard request and response language for accessing resources. Although XACML is widely accepted as a reference solution for access control, it does not provide any support to verify information within access requests, it just expects correct input.

The following study as a master degree project investigates major challenges and mechanisms to offer access control on the user's hand in a distributed social network environment (DOSN). Specifically, the main focus is to formulate an XACML-based privacy-enhanced access control mechanism for the subscribers of any DOSN.

1.1 Elementary Concept: OSN

The main goal of a peer-to-peer social network is to offer existing features of current OSNs while confirming privacy of the subscribers. The broad ranges of functionalities offered by the OSNs can be classified and defined as follows.

- **Profile:** The profile represents the identity of an individual user in an OSN. During the time of sign-up, the profile is generated and contains personal information, a list of social connections, resources, etc. that is everything the particular user wants to share through OSN.

    - **Social Connections:** A list of people such as friends or family to indicate existing social relationships of users. Almost all OSNs offer special functionalities so that a user can connect with new people based on different but related categories such as common interest, same school, etc.

    - **Resources:** Different kinds of resources such as pictures, videos, events, documents, groups, wall (space to broadcast messages to all or selected social connections, public messages), which are stored under specific user profile.

    - **Communication:** Since communication is the main motivation behind the development of online social networks, they offer different ways to communicate for the users.

        - ♦ **Public Message:** Interaction through a "wall" (like a noticeboard), which may be public or semipublic depending on users' settings.

[2] http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml

- **Private Message:** Same as existing electronic mail, only visible to the sender and receiver.

- **Instant Message:** Synchronous text-based messaging among users.

1.2 Background Information

In this section we describe recent research on using distributed infrastructure for forming social network service in a privacy-preserving way. Also in order to present the background concept and scope of this thesis, we discuss about different language-based security solutions to protect personal information in web-based applications. Therefore, we start by discussing progress of PeerSoN, one of the projects, which focuses on p2p infrastructure for social networks while preserving the privacy of the users. Again, as our target is to develop an XACML-based access control mechanism for distributed social networks, we provide a brief overview of XACML. Additionally, we discuss about review of some projects, where the core aims are either privacy issues of personal information, or XACML-based access control mechanism.

At the initial proposal, PeerSoN offers 2-tier p2p-based architecture for social networks where in one tier distributed hash table (DHT) is used for lookup services and another tier consists of peers to communicate and store users' profiles. At the beginning, the primary focus is to describe basic functionalities of OSN in p2p infrastructure to remove the dependency on a central authority (service provider). A message exchange protocol is also implemented to show how one peer communicates with another peer, and also with look-up services. This infrastructure strengthens the privacy settings. However, due to the absolute decentralized architecture, many research problems require careful consideration. Among these research problems, PeerSoN targets to solve these two questions: "How to offer a secured distributed storage mechanism to the users of DOSN" and "How can the subscribers control access to their resources?"

A number of p2p storage systems have been proposed in the last few years such as FreeNet [3], Collaborative File System (CFS) [4], PAST [5], OceanStore [6], PeerStore [7], pStore [8], Secure-Backup [9] but very few of them concentrate on the factor, "Availability of the Information" which is the most important concern in the perspective of social networks. Therefore in DOSN, the storage mechanism must ensure "maximum availability" as well as "complete privacy". Rammohan Narendula et al. [10] proposed a design for decentralized storage of OSN profiles by taking into consideration both "users' geographic locations" and their "online duration". But the architecture of the proposal considers of storing information as ordinary format in other's machine and as such increases the possibility of leakage of personal information.

PeerSoN already addressed this storage (replication) issue and offered a mechanism [1] to choose a group of peers to replicate each other's information while considering these factors: "maximum availability" and "minimum number of replicas". The suggested heuristic architecture performs well in simulation environments regardless of a selfish behavior of some peers. But the investigation is going on to offer a mechanism to check the reliability of the replication peers or a method to store securely on untrusted peers.

Additionally, PeerSoN suggested a method [11] to recover private keys or passwords in a p2p architecture. In the centralized system if any user forgets or loses his secret key, he/she can recover it by answering some private questions or by providing email address. But in a p2p system, secure backup, and recovery of secret key is an important and challenging problem. In PeerSoN a mechanism is offered to select the most trustworthy delegates to backup private keys and a threshold-based crypto-graphical protocol[3] is applied to ensure secure secret sharing on distributed social networks.

Moreover, EU project PRIME[4] (Privacy and Identity Management for Europe) focused on the problem of an identity management and thereby, identified two key elements [12] for managing identities : anonymous credential system; and a standard policy language, to support anonymous credentials, access control policies, data handling policies, regulation of communication between parties, policy negotiation and evaluation.

Furthermore, another EU project PRIMELIFE[5] (Privacy and Identity Management to Future Networks and Services) focused on privacy issues of personal information in Internet-based applications. The project also concentrated on data handling policies in two perspectives: users (complete control over their data) and organizations (who offer services but follow data handling preferences proposed by the users). As a part of the project PRIMELIFE result [13], it was elaborately described, “the use cases, requirements, and mechanisms for privacy enhancing access control services in social networking sites as well as in collaborative workspaces”.

In order to handle these privacy issues of users, OASIS[6] (Organization for the Advancement of Structured Information Standards) ratified XACML (eXtensible Access Control Markup Language) [14], a standard, platform and application independent policy language for presenting and exchanging access control policies. XACML is considered as a standard solution in the field of security because of its two important features: “expressiveness”, providing rich set of built-in functions to express policies based on both dynamic and static properties; and “extensibility”, offering the ability to accept new functionalities based on requirements of underlying applications. Moreover, XACML describes authorization request and response layout to make it suitable in large-scale environments.

On the other hand, Claudio A. Ardgana et al. [15] claim that XACML does not provide effective privacy and also the result of PRIME is not applicable in the real world applications because it requires many alterations in the legal system. Again, Claudio A. Ardgana et al. [16] identify that XACML does not support any credential-based access control mechanism; access control based on partial disclosure of attributes; and also access control based on fulfillment of conditions on specific attributes without revealing the actual values. Therefore, they proposed an architecture for credential-based access control while preserving the privacy such as anonymous credential by combining XACML with SAML [36] (Security Assertion markup language). SAML is an XML based language, standardized by OASIS for exchanging authentication and authorization data

[3] http://www.c2.com/cgi/wiki?ThresholdCryptography

[4] https://www.prime-project.eu/

[5] http://www.primelife.eu/

[6] http://www.oasis-open.org/home/index.php

among different security domains such as between an identity provider (a producer of assertions) and a service provider (a consumer of assertions). The “SAML profile of XACML” [41] offers to provide privacy-enhanced authentication and access control mechanism. This profile defines how to protect authorization requests and how to transport the XACML scheme, by combining XACML’s authorization policies expression and evaluation facilities with the SAML’s assertion administration abilities.

1.3 Research Problem

After defining a basic p2p-based architecture for social networks, now, the main focus of PeerSoN is to confirm the privacy of the users’ information. In order to achieve this, first step is to give full control over data to the users, and therefore, only authorized users can access resources. Although there are a large number of studies about access control mechanisms, there are very few mechanisms targeting the users of online social networks. This is the main inspiration for this thesis.

Generally, the subscribers of OSN are from different categories (different ages, different countries etc.) and their access control requirements spread over a broad dynamic range. Therefore, at present, the main focus of PeerSoN is to find a method or a policy language, to express this wide range of access control policies and to enforce these policies in a distributed environment. Target of this thesis is to achieve the following goals:

- Investigate the challenges of offering access control policies in a distributed environment.

- Develop a realistic access control mechanism to integrate with a distributed online social network such as PeerSoN.

- Evaluate the applicability of XACML as an access control policy language to express authorization policies of social network users and identify its extension points to make it applicable in DOSN.

1.4 Contribution

Our main target is to build a privacy-enhanced access control mechanism for the users of DOSN. Due to the wide ranges of requirements, at the beginning, we mainly focus on the expressiveness of authorization policies and finding a policy language to define these policies properly. The contribution of this thesis is outlined as follows:

- Our first contribution is an overview and discussion of current access control models and existing solutions and their lack of suitability in distributed social networks, and adaptation of rule-based access control for DOSN users. Further, the requirements for access control in a p2p architecture are detailed.

- Another contribution is the proposed profile structure for any user in p2p social network applications. The structure is provided not only to protect resources of the users but also to protect the privacy of the social connection lists. The structure is offered to store all the resources as separate blocks so that unauthorized users do not have any knowledge about the existence of unapproved resources. Moreover, the offered structure is such that users can define authorization policies based on social relationship or trust-level which is very common in real life relationships.

- A further contribution is the implementation of common access control policies in a standard, platform independent policy language, XACML and combination with a suitable authentication method (in SAML) to support privacy features.

- Finally, the analysis of the proposed architecture against different security threats and identification of weaknesses is another contribution of this thesis.

1.5 Structure of the Thesis

The report is organized as follows. **Chapter 2** illustrates existing access control models and solutions and their scope of applicability in the perspective of distributed social networks.

Chapter 3 provides an Overview of the requirements of p2p social networks and a description of a basic structure of users' profiles in social networks.

Chapter 4 presents the XACML and SAML languages and their architectures, and the integration of both and their extension points for social network users.

Chapter 5 describes details of the implementation of access control policies in OSN while considering both static and dynamic constraints.

Chapter 6 illustrates the rationales behind our design choices, followed by an evaluation against different security threats, and also possibility of alternative approaches.

Finally, **Chapter 7** presents the conclusions of this thesis by including the achievements and possibility of future research.

Chapter 2

Related Works and Applicability in DOSN

The goal of this chapter is to present current access control models and research findings, related to this thesis and to point out the limitations and scope of applicability, of these existing solutions in the perspective of distributed online social networks.

2.1 Existing Access Control Models

Recently, necessity of access control for social network contents, has received much attention. Some works have provided solution for offering access control mechanisms, to the users in online social networks. On the other hand, because of increasing popularity of content distribution in the p2p architectures, the security issues in distributed architectures are also taking attention of the researchers. However, providing fine-grained access control, in a distributed online social network is completely new and challenging research problem and so far, there is no complete solution to handle this.

William Tolone et al. [17] describe elaborately, the security issues in collaborative systems, such as the requirements of access control mechanisms in collaborative workspaces and the limitations of existing access control models against different threats. Existing access control models and the questions about their applicability, in online social networks are explained in the following sections.

2.1.1 Access Matrix Model (AMM)

Lampson et al. [25] present a model, to define the access privileges (different types of operations, such as read, write, etc.) on an object or resource to a subject. There are different ways to implement this model: *Access matrix,* where the users and resources are presented as distinct rows and columns and the values of specific row and column denote access rights; *Access control list (ACL),* where each resource is related to ACL to indicate, who have access permission on that resource, and what are the types of operations, such as read, write, modify, etc.; and the *capabilities,* where each subject is connected to a capability list to define his/her access rights on each resource.

Why AMM is not applicable in OSNs:

- With this model, it is not possible to offer advance access control policies, such as controlling access based on subject's attributes, such as age, trust, etc. Moreover, this model fails to support, single access control policy for a group of users, who have same attributes.

- It is difficult to manage dynamic access control in this model, such as revoking or changing access rights of users which are very common in OSN.

- Additionally, this model does not offer any method to express access rights based on properties of resources, such as title, version number, etc. and also based on context, such as time, date, etc.

- This model is dependent on underlying platform. Specifically, it is not possible to provide access to a resource over operating systems, where platform independence is one of the major requirements in OSN.

Although this model is inadequate to present wide ranges of access rights, the idea of separate access control for specific operation (read, write, modify, etc.) is valuable, for usage within an access control mechanism for DOSN.

2.1.2 Role-based Access Control (RBAC)

Sandhu et al. [26] propose a model, which offers access control based on roles (assigned to subjects) rather than specific identity. The roles are formed based on different criteria (depend on the requirements of the application and administrators), such as qualification, duties, etc.; and access rights on different resources are based on these roles. Finally, each subject is assigned a role, or multiple roles and his access rights depends on this assigned role.

Why RBAC is not applicable in OSNs:

- In OSN, it is not logical to control access on the basis of the roles only. Because access rights on the basis of resources' properties, require proper consideration, which is not possible to express by this model. Again in some cases, such as for defining access rights for a single user needs to assign a new role to that user, which is difficult to manage in a highly dynamic distributed environment.

- RBAC does not facilitate to express fine-grained access control on objects, such as if a user has access permission on the previous version of a document, but he is not allowed to access the current version.

This model alone is not enough to define access rights of the users in OSN; but definitely, the idea is suitable for expressing access rights in some cases, such as for groups, or events in OSN.

2.1.3 Task-based Access Control (TBAC)

Roshan K. Thomas et al. [27, 28] propose a model, which broadens the functionality of the traditional *subject/object-based access control,* by offering new access constraints based on the context (*task)*. In this model, access privileges depend on procession of any task. In each step of progress, there is a disjoint protection state, which encloses a set of authorizations and this authorization set is varying according to the progress and environment of the task.

Why TBAC is not applicable in OSNs:

- In OSNs, it is not always possible to control access of the users based on contextual information (work-progress). Definitely, it requires more features, such as defining access rights based on properties (users, resources).

- It is difficult to support the dynamic environment of OSNs, such as revoking, or changing access rights of users.

2.1.4 Team-based Access Control (TMAC)

Roshan K. Thomas et al. [29] presents a model, which offers access rights, to the group or team of users rather than to a single user. Forming teams, is more logical and common than creating different roles for dividing users. In this model, two important factors are considered for making groups or teams to associate with specific access rights: *User context,* to segregate users based on their current roles in a group; and *object context,* to make groups of resources with same types (based on applications or criteria).

Another model is proposed called "context-based TMAC (C-TMAC)" by Christos K. Georgiadis et al. [30], which has the functionality of both TMAC and RBAC and additionally, supports access control on the basis of contextual information such as time, date, etc.

Why TMAC and C-TMAC are not applicable in OSNs:

- Both of these models are based on standard RBAC model, but until now, there is no complete implementation of these models to make it clear, how will the concept *team* or *context* be integrated with RBAC.

- *C-TMAC* model seems to be very interesting for usage in scope of authorization control in OSNs. However, in some situations, such as providing same access rights to two different persons from two different teams are very cumbersome to express, by this model.

Although there are some limitations of these models –TMAC and C-TMAC, the core idea of these models is perfectly suitable for authorization control in OSNs. As for example, in OSNs, the access rights completely depend on relationship type. Therefore, it is a good idea to divide the users in teams or groups according to the basis of relationships, and then, define access control policies (different constraints) for those teams.

2.1.5 Spatial Access Control (SAC)

A model is proposed by Adrian Bullock et al. [31] with the aim of using in collaborative platforms. This model suggests keeping the security methods hidden from the users. The architecture of this model is the combination of two main modules: *a boundary,* to split the whole application environment into small regions and to control access, within these regions by the concepts of *credentials*; and *an access graph,* to present conditions for access control and to limit the movement within application environment.

Why SAC is not applicable in OSNs:

- This model has failed to offer fine-grained access control, which is one of the major requirements for the users of OSNs.

- It is very difficult to divide an online social network in regions, or outline boundaries. That's why; the applicability of this model in OSN is a challenging problem.

2.2 Earlier Proposals for Access control in OSN

B. Carminati et al. [18] introduce rule-based access control mechanism for the users in online social networks, where the policies are stated as conditions based on *type (relationship type)*, *depth (relation depth)*, and *trust* (between any user and his/her social links). The authorization policies are confirmed by the subscribers, according to the rule-based approach. When a subject requests to access a resource, he needs to submit a proof that he has the necessary privileges to access that resource. As the access rules are based on *relationship* (between requester and owner), the requestor needs to submit a certificate to authenticate his/her relationship (type, depth etc.) to the resource owner. The *relationship certificates* are encrypted with symmetric key (known to the storage nodes and relating people), and stored in a central node.

This is a very flexible access control mechanism for the users to control access to their resources, as anybody can access resources without being socially connected with the resource owner. The central node is trusted storage provider of certificates for all nodes in the network, and therefore, it can create and submit certificate to the resource owner, to show the relationship between the requester and the owner. However, a main problem of using it in a DOSN is the dependency on the central server, for relationship authentication. Further, another question arises - is it logical to provide access to the resources to whom the resource owner is not connected in the social network? Another, very important factor is that there is no clearly presented method, for revocation of certificates, that is, how to manage changes in trust level among relations, which is very common in social relationships.

J. Domingo-Ferrer et al. [19] propose a way, to keep the privacy of relationships in OSNs, by using public key cryptography. And show that, using public key instead of symmetric key for cryptography helps to decrease traffic overhead, which occurs in private relationships. They offer two improved extensions of the mechanism, presented by B. Carminati et al. [18]. First, removing dependency on a central node, and second, using public key instead of symmetric key, to encrypt the relationship certificates. The authors use the same framework of the B. Carminati et al. [18], but define access control rules based on *trust* and *depth* of the relationship links, between the requestor and the resource owner. Also the authorization decision imposed is based on the reachable relationship paths between them. In this method, no central node is required to store certificates or compute the *trust level*, as the owner connects with the related nodes and compute *trust* (of the requesters) based on the received certificates for those nodes.

The offered protocol achieves the protection of private relationships in OSN, while giving the advantage of being fault tolerant. It is also free from depending on trusted third parties. However, this framework fails to represent access control rules based on the properties (static or dynamic) of the users or resources, and is also based on complex conditions. Further, this mechanism is not sufficient to present fine-grained access controls for a single resource, and also unable to specify exact operations types in rules, such as offering only *read* or *write* access to a single resource.

F. Beato et al. [20] identify the problem of depending on OSN providers, to enforce access control policies on users' information; and provide a mechanism, such that users can control access to their own information and resources by themselves. The proposed architecture is platform-independent that provides a method to define access control rules for selective audiences, and uses encryption technique to impose these rules. The authors propose a tree like structure for the users' profiles, by diving into two main classes: *connection class,* for maintaining the social links of a user; *content class,* for maintaining information and resources of that user. These two main classes can be subdivided into subclasses as per requirements, such as friends, family, or work, hobbies, etc. The proposed mechanism uses role-based access control, by linking the classes with associated access rights. When a user wants to share a new resource, he just connect the resource with the content class, and a hybrid encryption technique (encrypting the content with a random symmetric key and then symmetric key is encrypted with all the eligible users' public keys) is used to enforce authorization policies.

The idea of providing access control to the group of users in OSN is very efficient, but not sufficient to provide extensive access control policies to all the users. Again, one question arises - if it is required to provide access rights to one from friend class and another one from family class, then how will this mechanism define the authorization policy? Furthermore, in this proposal, revoking access rights of a particular user, requires changes in the symmetric key of all from the authorized content classes (revoked user), which is very costly in a highly dynamic environment.

Randy Baden et al. [32] propose architecture, called "persona", to give full access control on the resources to the users, in OSNs, by taking advantages of encryption based on both attribute and public key. The assumption is that all the users in *persona* are identified by their public keys, and they exchange these keys with their social links out of band (by telephone, email, direct meeting, etc.). The basic idea of the *attribute based encryption (ABE)* is to encrypt resources based on the attributes. Therefore, the users, who have these specific attributes, are eligible to access the resources. The attributes can be presented, as string, or as a numeric value for comparison. ABE is used in "persona" for audience segregation, to divide the users to different groups, according to similarity of their attributes. Then the ABE secret keys are distributed to the correlated users, by encrypting it with their individual public keys. The Symmetric key is used to encrypt resources of the users, and then the symmetric key is encrypted with the ABE secret key.

The authors identify the same problem, proposed by F. Beato et al. [20], that is, absolute dependency of the users on the OSN providers, to protect their resources; and propose the architecture based on cryptography. The proposed solution considers all the functionalities of the current social networks and provides a flexible user-defined access control policies. However, sometimes, it is required to give access control, to some specific users from a group or from different groups rather than the whole group. Then it becomes burdensome in persona. Because in this situation, either the owner has to create a new group, or modify the existing group. Most importantly, in persona a user needs to maintain a large number of keys to access resources, such as one symmetric key and one group key for a single resource, and also using re-keying for revocation of the access rights, raises the number of keys for a single user. Finally, the architecture is proposed for centralized architecture. Therefore, it requires further evaluation, to make it applicable in decentralized architecture.

Again on this same issue, another proposal is presented by Jonathan Anderson et al. [33] for client server architecture, based on cryptography. The architecture is not only protecting users' resources (contents and relations) from OSN operators, but also prevents users, to gain knowledge about unapproved resources. The architecture consists of four layers –*Application layer, Data structure layer, Cryptographic layer* and *Network layer*. In *application layer,* all the applications are proposed to run inside a secure sandbox, to limit access to users' private information. In *data structure layer,* the content is divided among blocks, and maintain a tree like structure to prevent other users getting information about existence of the resources beyond their access limits. The *cryptographic layer* offers: *confidentiality,* to guard resources (both content and links) from unauthorized access, and *integrity,* to validate authenticity of the resources. As the proposal is for client server architecture (clients store their blocks of resources to the server), therefore, in *network layer* only two commands are executed –*GET,* to access a block of a resource from the server; *POST,* to update an existing block, or to provide a new block to the server.

The suggested architecture is mainly focused on, how to enforce access control on the resources in existing OSNs, having an assumption that servers (service providers) are untrusted. The architecture does not offer any method or model to express access control policies for the users. Moreover, no clear idea is presented regarding revocation of access rights from a user or group of users.

2.3 Previous Research in Authorization Control in P2P Architecture

The security issues in p2p architecture are becoming more significant with the adoption of the p2p-based applications. But offering an effective access control mechanism on resources, in a p2p network is still challenging, due to its complete decentralization, and lack of any central authority, or trusted third party. M. S´anchez-Artigas et al. [21] are motivated by this identification, and propose an access control enforcement protocol "pace", to provide privacy to both access control policies and access privileges. This protocol engages three different types of participants: *Object Owner***,** responsible for taking decisions about access to the resources. *Policy holders* are responsible for determining authorization decisions on behalf of the resource owner (by storing the authorization policies), during the period when the owner is in offline. And *Storage nodes* are responsible for storing the resources in encrypted form (encrypted with symmetric key). When a requester sends access request to a resource, before getting access permission, he needs to submit authorization certificates (as a proof of eligibility) to the policy holders, which he previously acquired from the resource owner. When the requester's authorization is granted, he gets the decryption key to access that resource from the storage nodes. In "pace"**,** hierarchical access control model is used to minimize the number of keys for the resource owners. For authentication, each user maintains a trusted group (TG) of users, to perform the challenge-response authentication on his behalf.

The proposed protocol for access control, with the aim of using in p2p collaborative systems is successful, as it is free from centralized control. Also it is not dependent to any trusted third parties. But for distributed online social networks, the proposal is not adequate for defining logic-based access control policies to the users, as the proposal does not focus on expression of common constraints of authorization policies, such as attributes of requesters, replicas, context, etc.

C. Sturm et al. [22] suggest a fine-grained access control mechanism for p2p collaboration, based on local access control components of the participants, where access control policies of every peer transfers to each other using *XACML*. Two methods are proposed for combining these local policies; both of these methods are evaluated against different performance criteria, and their comparative benefits and drawbacks are depicted. One of these methods is based on linking different types of policies after exportation from local peers. Another one has its grounds on installing a distributed access control directory, to locate different policies. The overall idea is to give an opportunity to the peers in a p2p network, to create a universal access control element, without any central authority. And the participant peers can either control access to their resources completely, by themselves or select a number of delegates (other peers based on some criteria), to manage access control.

This proposal is unique approach for p2p-based access control, and includes *XACML* for global access control model, which makes it more platforms unaffiliated, dominant, and flexible. However, its applicability in DOSN is uncertain, due to different people from diverse sectors, with dissimilar types of platforms that are the subscribers of online social networks. And therefore, it is difficult to assume that everybody would have their own access control component and that will be using in an efficient way.

E. Palomar et al. [23] offer a certificate-based access control mechanism for p2p networks, where digital certificates are used to determine authorization decisions. Each peer of the network categorizes its content according to different security levels, and other peers, who want to access those contents, are required to meet that level. The content or resource (m) is encrypted with a symmetric key. The certificate of a resource (Cm) is used, to represent the authenticity of the access request, and enclose the security clearance to access that resource.

Offered mechanism is well suited for file sharing applications or collaborative environments, but many factors need to be considered properly, in order to use it in a social network environment. Such as what are the basis of the certificates; how to provide a certificate based on the properties of objects, rather than subjects; if a user maintains a large number of social relationships, then how many certificates he needs, to access different resources from different relationships; how many encryption keys, users require to store. Additionally, changing symmetric keys in such dynamic environment raises many performance questions.

One of the major problems in a p2p architecture, "providing and maintaining access control to replicated resources, where the replicas are not trusted equally", is identified by T. Wobber et al. [24]. A mechanism is offered to express, and to store fine-grained consistent access control policies in a number of replicas. The mechanism suggests how to select delegates (as authority), to set new policies. The system uses *SecPAL (Security Policy Assertion Language)* for expressing security policies and provides a mechanism, for ensuring ultimate consistency of access control polices, in all the replicas. Furthermore, the proposed implementation uses a reference monitor to enforce access control policies at the execution time.

In this proposed architecture, one of the trickiest problems in p2p networks, "*consistency of access control on replicated resources*", is handled efficiently. But the authors predominantly highlight on maintaining the consistency of access control rules in all the replicas, rather than expressing authorization policies based on different constraints.

Youssef et al. [51] mainly, focuses on access control mechanism in p2p OSN through encryption. He depicts major challenges to offer an access control mechanism in a distributed architecture, and proposes a hybrid encryption technique to prevent unauthorized access on users' resources. He suggests creation of a unique key (hash of the combination of public key and file name), for each uploaded file, and to store the associated decryption key in a key list object, which is signed by the owner's private key. The decryption key is encrypted with symmetric keys and then, that key is encrypted again with authorized users' public keys. For membership revocation, the objects are encrypted with new keys as well as the key object list is updated.

The proposal can be used as an initial step in offering privacy to the users of DOSN. Using encryption technique in DOSN is an effective way to enforce access control policies, and to give control on the owners' hands over their resources. But the proposal does not offer any method or model to express wide ranges authorization policies. Moreover, to make it suitable for distributed environments, some other factors require further consideration, such as – key management issues, performance issues, and especially, evaluation of the whole system against different security threats.

Chapter 3

Requirements Overview

This chapter provides an overview of requirements that should be considered, to offer privacy-aware access control mechanism in p2p social networks. Based on these requirements, a profile structure for OSN users is described, in such manner that they can control access to their resources based on social relationships.

3.1 Requirements for Privacy-enhanced Access Control in OSN

In OSN, access control model provides solution, who and how can one access sensitive private information and recourses, of the subscribers. In EU project PRIMELIFE, one of the documents titled "Requirements and concepts for privacy-enhancing access control in social networks and collaborative workspaces" [13] presents the use case scenarios, detailed description of the requirements and necessary mechanisms, to offer privacy-aware access control service in online social networks and collaborative workspaces. Although access control requirements in OSN vary due to the diversity of subscribers, we can summarize the prerequisites of access control mechanism in OSN as follows:

- The access control mechanism should support selective access control, to share selected piece of resources with selected group of users.

- The access control mechanism should offer proper access control to the profile data, to make sure that this information is not used anywhere else without knowledge of the owner.

- The access control mechanism controls not only unauthorized access but also prevents unauthorized downloading of any information.

- The authorization mechanism should offer user-friendly interface for the users to define their access control rules, and also the way to enforce these access control rules.

- The mechanism should generate notifications and informs the owner, about usage of their resources.

- The users should have appropriate access control to offer privacy, for both resources (uploaded content and information) and relationships (social connections).

- The mechanism should support, expressing access control policies based on the identifiers, roles, groups, properties or attributes, context, etc.

- The mechanism should have a way to define access control constrains based on complex rules or logics.

- The authorization mechanism should support fine grained access control policies for resources.
- The mechanism should facilitate users to define access control policies not only based on predefined constant properties but also, based on dynamic properties of the resources, or owners, or requestors.
- The mechanism should offer options for users that allow specific operations on resources, such as read/ view or write/modify.
- The mechanism should facilitate methods for revocation, to withdraw or modify access rights for a specific user, or group of users.
- The mechanism should have a way, to define single policy to control access on a group of resources.
- In general, the access control mechanism should be transparent for the authorized users.

3.2 Requirements Analysis for P2P OSN

The shift from centralized to peer-to-peer infrastructure for social networks add some new requirements for authorization control. As there is no central authority (OSN provider) for access control mechanism, we need to think about both "policy expression" and their "enforcement". Considering both of these issues, we describe an approximate list of the requirements for a user-controlled access control mechanism in p2p OSNs.

- Provide access control to the resources without any central authority or trusted third parties.
- Provide access control for both users' resources (private information, pictures, videos, etc.), and social connections (relationship information).
- Provide such access control that users can define specific operation types on a resource. (access permission only for "read" or "write" operation)
- Provide access control to a group of users on a group of resources, that is, users can define a single policy for a group of users, on multiple resources.
  Access control based on group/Team: *Divide all social connections to specific groups based on properties, such as relation type, trust, etc. Also different resources are combined and grouped based on properties, and then authorization is offered to this group of users.*
- Provide access control to different users from different groups on a single, or multiple resources.
- Provide fine-grained access control to the users on resources, such as different access control policies on different versions, of the same document.

- Provide access control based on roles.
  Access control based on role: *In social network groups, assign roles to the users on basis of some criteria, such as membership type, qualification, etc. and offer access control to those roles.*

- Provide access control to a user on a single, or multiple resources.
  Access control based on identifier: *Provide access control to a user based on his identifier, such as email address, or user name in the OSN.*

- Provide access control based on attributes of the users or resources.
  Access control based on attributes: *In OSN, define access control policies based on users' attributes, such as trust level, age, profession, etc. or resources' properties, such as type, version number, creation date etc.*

- Provide access control based on context.
  Access control based on context: *In OSN, set access control constraints based on context, such as access time or dates.*

- Revoke access control from a user or group of users (withdraw, or modify access rights of any user, or group, or role).

- The access control policies should be consistent throughout the whole system.

As mentioned before, access control requirements in OSN, spread over a broad dynamic range, due to the diversity of their subscribers. Further, in the previous chapter, we discussed about lack of suitability of recent access control models in DOSN. Therefore, based on these observations, we adopt rule-based access control mechanism for p2p SN, where the authorization conditions are based on groups/teams, or role, or attribute, or identifier, or context, or combination of all of these factors.

After selecting an access control model, next question is - how to enforce these access control policies. Mainly, there are two possible answers:

- Using an encryption technique (Symmetric, or Asymmetric, or Hybrid).

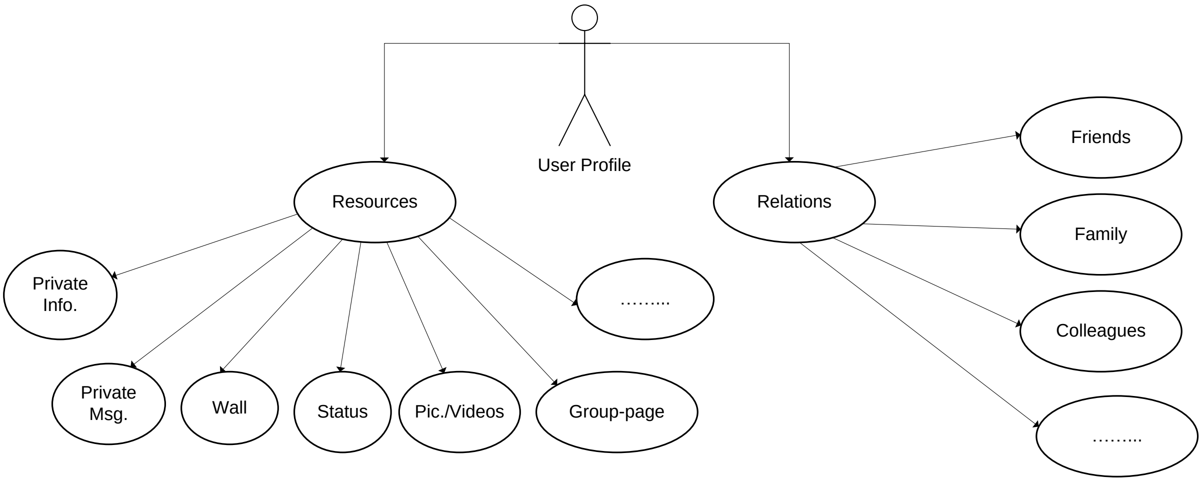
- Using software based enforcement technique (Browser extension, or web server, or reference monitor).

3.3 Basic structure of OSN Profiles

In OSN, it is required to offer privacy for both uploaded resources and relationships information. It is normal that, in real life, all relationships are not in the same level and also not mutual. Therefore, disclosing how much information is hidden from a user, strongly affects his social relationship. In order to prevent users, from gaining knowledge regarding unauthorized content, we offer a profile structure. The whole profile is divided into small blocks, and connected with each other like a tree. Moreover, in social relationships, access privileges mostly depend on “relation type” and “trust level”. Thus, our proposed structure facilitates users to define access control policies based on these features while including other factors pointed out in requirements list. The overall structure of users’ profiles in OSN with different possible access types is presented in figure 3.1.

A brief overview of our proposed structure is presented below:

- Social relations of the users are maintained by including three attributes – “unique identifier/pseudonym”, “relation-type” (friend, relative, colleague, etc.) and “trust level” (High, Medium, Low).
- The *profiles* of the users are divided into two main categories- *Resources* and *Relations*.
- The social links of a user are divided into groups based on the *relation-type* attribute and are stored under the category, *Relations*.



ACCESS CATEGORIES

- Who can send and nobody except the receiver & sender can access? (Private Msg.)
- Only available for View/Read access to whom. (Private Info.)
- Who has read/view and who has write/comment access? (Status, Wall, pic/videos, group-page)
- Who has tag access? (Pic/videos)
- Who has administrative access? (Group-page)

Figure 3.1. Basic structure of OSN user profile.

- The category, *Resources* is also divided into groups based on types, such as private information (Profession, age, educational qualification, family information, etc.) private messaging, public messaging (wall, group page, status, events, etc.), pictures or videos, etc.

- Access control constraints on resource or group of resources are defined, to specify specific operation type, such as read, or write on the *wall*, or *status* (public messaging); view, or comment, or tag on pictures, or videos; or read, write, administration on group-pages or events.

Chapter 4

Language-based Privacy

This chapter presents an overview of language-based privacy for the users of OSN. We start our discussion from the basics of an authorization policy language, eXtensible Access Control Markup Language (XACML) and then, suggest two extensions of XACML to make it applicable in DOSN. After that, we provide the details of Security Assertion Markup Language (SAML) and its integration with XAML. At the end of the chapter, we point out some major criteria, for selecting a suitable authentication method for OSN users.

4.1 Rule-based Access Control Using XACML

Although a large number of languages have been proposed so far for expressing access control constraints, XACML has gained the most attention because of its standardization in the field of security. It also offers a standard request and response language for accessing resources. Again, the ability of expressing access control constraints on the basis of dynamic properties of the owner, or resources, or requesters, rather than on fixed values, makes XACML more acceptable in real world. Last but not the least, two most important features of XACML are, "flexibility", and "extensibility", which make it application independent. By "flexibility", it provides the capability to detect policies from distributed locations, and by "extensibility", it offers the ability to accept new features, on the basis of application specific requirements.

4.1.1 eXtensible Access Control Markup Language

XACML [14] is an OASIS standard language based on XML, for expressing fine-grained authorization policies to protect resources, and also to handle access requests and corresponding responses for these resources. Moreover, XACML offers the way to evaluate corresponding policies for an access request, and generate subsequent response, which contains the decision as *Permit*, or *Deny*, or *Intermediate* (Decision cannot made because of missing any required value or any types of errors), or *NotApplicable* (Corresponding policy is not found). Further, it offers a basic architecture, combination of some components, which are probably necessary to develop a complete solution for an access control mechanism in real life applications.

4.1.1.1 Core Features

XACML mainly, provides an access control and authorization mechanism, to express various access control constraints; to collect related data to evaluate policies; to evaluate these policies, and to return the decision based on the evaluation.

The core functionalities supported by standard XACML (version 2.0) are summarized as follows.

- **Expressing policies based on attributes**: XACML offers the advantage of presenting access control policies based on properties of the users (e.g. name, email address, age etc.), or resources (e.g. title, version etc.), or context (e.g. time, date etc.). XACML also suggests some standard operators and functions to calculate, or equate different types of attributes values, and also offers extension facility to include new operators, or functions.

- **Expressing Policies based on multiple subjects and multi-valued attributes**: XACML supports defining single authorization policy for multiple subjects, and also has the features to express policies based on an attribute (of a subject, or resource), which has multiple values.

- **Distribution of Policies**: XACML offers various features for the policy makers to describe a single policy by different parties, with enforcing it at different enforcement points. In addition, XACML provides the facility to reference one policy within another, where two policies may be stored in two different locations.

- **Rich set of Standard data-types:** XACML has a large set of built-in data-types as well as the facility to add new ones. It supports not only the primitive data-types from XML, such as *String*, *Boolean, time,* etc. but also some distinctive data-types, such as *rfc822Name, x500Name, yearMonthDuration,* etc.

- **Rule and Policy Combination**: XACML has the feature to combine multiple independent policies, or rules, and generate single decision (based on a combining algorithm) to control access to a resource. Therefore, all the applicable rules and policies are evaluated and a decision is generated, to handle access requests for that resource.

- **Combining Algorithms**: XACML defines both positive and negative authorizations, and also supports combination of several policies and rules, within single policy. As evaluation of the different policies generate different decisions, and therefore, to result one decision from these multiple decisions four different types of combining algorithms (in version 2) are proposed.

  Standard combining algorithms are defined as follows:

  - *Deny-overrides*: The final access decision is *deny,* if any single policy, or rule returns deny.
  - *Permit-overrides*: The final access decision is *permit,* if any single policy, or rule returns deny.
  - *First-applicable*: The final result is same as the evaluation result of the first rule, or policy, or policy set.
  - *Only-one-applicable*: This algorithm is suitable, where only one policy, or policy set is applicable, and the final decision is made based on evaluation of that single policy, or policy set. If more than one policy is applicable, the result returns *Indeterminate,* and if no policy is figured out, the result returns *NotApplicable*.

- **Domain independence**: XACML offers a way; therefore, the policy makers are completely dissociated from its architecture developers, such that XACML access control policies work in a consistent way, without any dependency on a specific implementation.

- **Obligations or Advices**: Beside authorization policies, XACML offers some extra constraints to control access, which are either mandatory (*obligations*), or recommended (*advices*). These additional actions required to be completed before, or after getting access permission.

4.1.1.2 Architecture

XACML defines four main components to define and evaluate access control policies, and to generate authorization decision, based on these evaluations. These components are: *Policy Administration Point (PAP)*, describes suitable policies by defining conditions for access control to the resources, and stores these policies in a suitable repository; *Policy Enforcement Point (PEP)*, accepts access requests and returns corresponding authorization decisions (also enforces these decisions), and also makes sure, *Obligations and advices* (if any) are satisfied; *Policy Decision Point (PDP)*, evaluates applicable policies based on the request, and generates access decision; *Policy Information Point (PIP)*, supports with the information about subjects, or resources, or environment to the PDP, which are required to evaluate policies.

Figure 4.1 depicts the whole work flow of XACML, starts from accepting an access request from a requester, and then, evaluates the corresponding policies, and finally, sends the response (access decision), and enforces it to the requester. Step by step work-flow [14] is descried below:

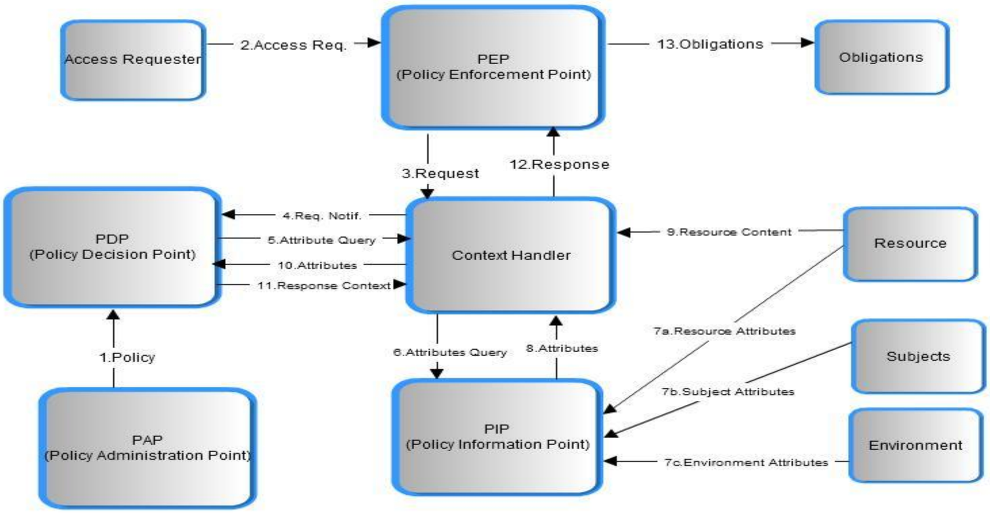


Figure 4.1. Overview of XACML data flow model (adapted from [14]).

Step-1. The access control policies are defined, and maintained in the *PAP,* and then make them available to the *PDP*.

Step-2. The requester sends a *request* to the *PEP,* to access a resource, or a set of resources.

Step-3. The *PEP* forwards this *request* to the *context handler* in application-dependent request format.
Step-4. The *Context Handler* translates the application-dependent request into *canonical format (XACML request context),* and directs to the *PDP*.
Step-5-10. The *PDP* detects suitable policies (depend on information within request) from the *PAP,* and obtain the missing attributes (within *request*), for decision making from the *PIP* through *context handler*.
Step-11. The *PDP* evaluates polices based on the values, obtained from the *PIP,* and policies from the *PAP,* and returns the authorization decision (as *XACML response context*) to the *context handler*.
Step-12. The *context handler* again transforms the *XACML response context* into application-dependent response (depend on *PEP*), and returns to the *PEP*.
Step-13. Finally, the *PEP* checks for completion of *obligations* (if any), and based on the access decision returned by *PDP,* enforces authorization decisions.

4.1.1.3 Policy Structure

The policy language model of the XACML is represented in Figure 4.2. The major elements of the model are:

1. ***Rule***: The most fundamental component of an access control *policy,* and expressed either independently, or encapsulated within a *policy* for exchanging *rules* among different actors in the XACML domain. *Rules* are checked based on its elements, and the main elements are:

    - *A target*: Defines single, or a set of resources, and subjects, and/or actions, to which the *Rule* is applicable. *Conditions* may be included inside it for additional refinement of its applicability.
    - *An effect*: Returns the result (either *Permit* or *Deny*), when the evaluation of the rule is true.
    - *A condition*: Enhances the scope of the *rule* by adding some constraints, as Boolean expressions.
    - *Obligation expressions*: Defines additional obligatory actions, which must be performed by the requester to get access permission for a particular resource.
    - *Advice expressions*: Defines additional recommended actions, which may be performed by the requester to get access permission for a particular resource.

2. ***Policy***: A *policy* is a set of *rules,* which are defined for access control to a single resource, or multiple resources. A *policy* contains five elements: *A target, A rule-combining algorithm-identifier, A set of rules, Obligation expressions* and *Advice expressions*. Except the *rule combining algorithm,* all other components are same as used within a *rule*. The *rule combining algorithm* defines the procedure, about how the results of the component *rules* are combined to generate a single authorization decision for the *policy*.

3. ***Policy Set***: A *policy set* is a set of *policies,* where *policies* are combined for authorization control to a resource, or multiple resources. A *policy set* also holds five elements: *A target, A policy-combining algorithm-identifier, A set of policies, Obligation* and *Advice expressions.*

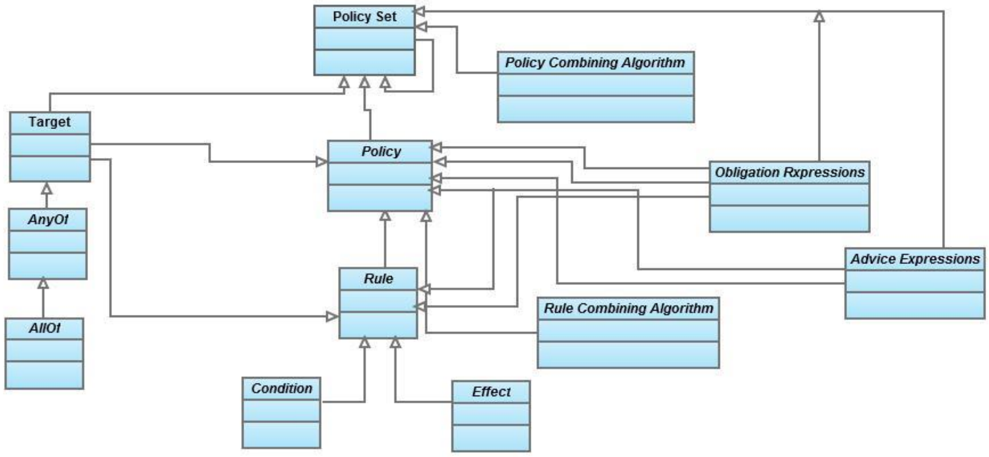


Figure 4.2. Policy language model of XACML (adapted from [14]).

4. ***Context***: A *Context* is the canonical representation of *request* and *response* to carry access request and authorization decision in XACML. As XACML is designed for a variety of application environments, therefore, the core architecture is isolated from its application environment. Thus, a *context handler* is used between *PEP* and *PDP* to transform the format of the *request* and *response* form domain-specific layout to XACML *context,* and vice versa.

   - *Request Context*: It contains information about the *Subject* (requester's properties, such as email address, unique identity, etc.), information about the *resource* (properties of the resources, such as resource ID, resource name, etc.), information about *action* (types of operations, such as read, write, edit, etc.) and information about *environment* (properties of the environments, such as current time, date, etc.).
   - *Response Context*: It contains information about the *resource* or *resources* (id of the corresponding resource/resources), *decision* about result of evaluation of the policies (*permit,* or *deny,* or *NotApplicaple,* or *Indeterminate*), *status* information (ok, missing attribute, etc.), and *obligations* (additional actions to get access permission).

4.1.1.4 XACML Profiles

XACML has offered different types of profiles, such as Digital Signature, Hierarchical Resource, LDAP, Privacy, Role Based Access Control (RBAC), SAML Integration, etc. for expressing access control policies in different applications. Among these diverse types of profiles Hierarchical Resource profile and RBAC profile are described in details below (From our point of view, these two profiles are required to express polices in OSN).

4.1.1.4.1 Hierarchical Resource Profile

XACML describes Hierarchical resource profile [47] to the express policies for access control to hierarchically stored resources (such as files, XML document, etc.). In this proposed profile, all the hierarchical resources are stored, either as a tree with single origin node or as a forest with multiple roots, but do not contain any circle with their connections. Every node in the tree or forest is considered as an independent resource.

The main feature of this profile is, controlling access for multiple resources by a single policy, and getting authorization decision for multiple resources, by submitting single request. To express the policies such that it is applicable for hierarchically stored resources, the nodes (independent resources in hierarchical structure) are presented in a consistent way to clearly differentiate between the parent and child node (relation among nodes). Again, XACML request context is necessary to express in correct form, to evaluate the corresponding policies perfectly, and to confirm authorized access, and also to deny unauthorized access. Further, for presenting access control policies for the hierarchical resources, XACML (version 2.0) supports four additional attributes:

- ***Document id****:* Define identifiers for the resources, which are structured as a hierarchy.

- ***Resource-parent***: Specify the resource, which is structured as a parent of any resource (requested one) in a hierarchical structure.

- ***Resource-ancestor***: Specify the resource, which is structured as child of any resource (requested one) in a hierarchical structure.

- ***Resource-ancestor-or-self***: Specify any resource (requested one), or its child resource in a hierarchical structure.

4.1.1.4.2 Role Based Access Control (RBAC)

XACML offers RBAC profile [48] to support core and Hierarchical role-based access control. This profile provides, how to assign a subject to a specific role or multiple roles, and also defines how to assign specific permissions based on these roles. Further, for supporting hierarchical role-based access control the relationships between roles are also considered, such as *senior, junior,* etc. In this profile four different, but related types of policies are described.

- ***Role PolicySet or RPS****:* Set of policies to relate a specific subject (have a specific role as attribute) with *Permission PolicySet that* includes all the access rights for a specific role.

- ***Permission PolicySet or PPS****:* Set of policies to describe all authorization privileges for a certain role, that is, to define access limits for each individual role.

- ***Separation of Duty PolicySet****:* Optional Policy set proposed in RBAC profile, defining constraints to bounds the number of roles for a particular subject.

- ***Role Assignment Policy or PolicySet****:* policy, or policy set to determine the relationship between a subject (Or more than one subject) and a role, or multiple roles, and also describe some conditions, which need to satisfy by a subject to have access rights (as a particular role). These types of policies are specially, used by the external entities to specify the role attributes for users.

According to the standard RBAC, it should contain five elements, and all of these are expressed in XACML as follows:

- *Users*: Define individuals, who are going to access the resources and represent as *subject* in XACML.
- *Roles*: Define *roles* based on environments, and in XACML expressed as *attributes* of *subjects.*
- *Objects*: Delineate the *objects* same as the XACML *resources*, for which the access control policies are defined.
- *Operations*: Same as the XACML *actions*, define the types of operations on resources.
- *Permissions*: Define the access rights based on roles, and in XACML these privileges are presented by two types of policies, *Role PolicySet* and *Permission PolicySet*.

4.1.1.5 Recent proposals of Extensions to XACML

As the existing architecture of the XACML is generic, therefore, some extensions have offered to add various new features. Two recent proposals to extend XACML are described elaborately, as they are more suitable in the perspective of OSNs.

4.1.1.5.1 Credential-based Access Control

J. Camenisch et al. [34] motivate from some real world authentication scenarios, such as - using national identity card, or driving license, to prove citizen's identity; using movie ticket to watch movies in theaters, or using membership card to get into a gym; and propose a concept of privacy-enhanced credential-based authentication and access control to the resources. Additionally, they offer an extension to XACML to support this feature. As most of the credential certificates contain private information, therefore, they proposed such that the service providers can inform requesters, which attributes are required to reveal for getting access. The authors identify that XACML is unable to express some important features, such as - submit access requests with credential certificates (complete set of attributes), submit a specific set of attributes from a certificate rather than all attributes bundle, and clear distinction between revealing exact value of the attributes and satisfying conditions on attributes. By taking into consideration all these limitations, the proposed extension offers to express the policies without any dependency on specific technology. The extension supports the features: *proof of ownership,* for binding authentication information with the credential certificate; *Selective attribute discloser,* allows to reveal only required set of attributes from a credential certificate; *Proving condition on attributes,* allows to prove conditions on attributes without revealing exact value of those attributes; *Disclosing required attributes*; *Signing Statements*; *Limited spending,* offers an upper boundary for using the same certificate for a particular resource. To support these features in XACML, the offer includes some additional elements inside the XACML standard *Rule*.

The proposed additional elements are:

- ***Own***: Define the credential id to bind some specific attributes with its owner with information about the issuing authority.

- ***Reveal***: Define, which attributes within a certificate need to disclose, or to satisfy some conditions over attributes values, without revealing the genuine values to get access permission.

- ***Sign***: An agreement about policies for using a resource, signed by the requester before getting access permit.

- ***Spend***: Define the maximum limit of using the same certificate to access a resource.

Further, to make XACML privacy-friendly, that is, to support the features of informing requesters about revealing specific attributes to access a resource, the extension offers to utilize the optional element of the standard XACML response context, *StatusMessage*.

4.1.1.5.2 PrimeLife Policy Language

In EU project PrimeLife, the researchers identified one of the major problems of the existing policy languages (such as XACML, P3P, EPAL, etc.), unable to enforce access control policies by technical measures and Claudio A. Ardagna et al. [35] offer to extend XACML to handle this problem. The proposal is to modify, or update the current structure of the *obligation* element in XACML, by adding some useful features - *temporal constrains*, *pre-obligations*, *conditional obligations*, and *repeating obligations* (an authorization system, define access control rules to handle, or distribute private information, which are already disclosed during the time of getting access permits, or which are required to reveal for access permission). Mainly, the extension is focused to include the data handling and credential capabilities in XACML. The XACML standard *Rule* structure is modified by adding some additional elements.

The additional elements are:

- ***Credential requirements***: Define the credential information to allow access permission for resources.

- ***Provisional actions***: Define pre-performed tasks to get access permission for resources.

- ***Data handling policies***: Define the rules; consist of a set of authorizations, and a set of obligations, to handle requesters' information, which they revealed during the time of access requests.

- ***Data handling preferences***: Define preferences of users as rules to show, how their information will be handled, when they get access permission. Like *data handling policies data handling preferences* are also expressed by means of a set of authorizations, and obligations.

The most important idea presented in this paper is to attach the *sticky policies* (set of granted authorization and promised obligations) with a resource, to enforce the security policies. The idea of the *sticky polices* is familiarized by Marco Casassa Mont et al. [46], where they present strong association between information and its access control policies so that the *sticky policies* decide access decisions for associated data and enforce privacy. Although the idea is very efficient to prevent unauthorized access to users' profiles in OSN, needs a concrete underlying hardware and software design to make it applicable.

4.1.1.6 Applicability of XACML in DOSN

Based on the requirements of an access control mechanism in DOSN and also for diversity of subscribers in OSN, XACML is the most suitable selection because of its following capabilities:

- Ability to express authorization policies with large complexity (different types of rules).
- Ability to detect security policies in a distributed environment.
- Ability to define policies based on the attributes of multiple subjects, or properties of resources.
- Applicability in diverse environments, with different types of resources, subjects, actions, policy locations, or policy combination.
- Specially, the ability to extend functions, or attributes, or data types, or policy combining algorithms, according to application specific requirements.

Although XACML provides a complete architecture, how to deduce the policies to a decision for a specific request, it does not tell any mechanism about implementation of the *PEP*, *PDP*, *context handler*, *PAP* or *attribute authority,* that is, lack of a privacy-enhancing mechanism. Some possible types of threats are identified in the OASIS specification document [14], such as unauthorized disclosure; unwanted message reply, insertion, deletion and modification. Further, there is no mechanism is suggested to ensure the authentication of both the requesters and the components, in the XACML architecture. Furthermore, no method is mentioned to offer privacy to the access control policies, such as confidentiality and integrity of access control policies (except resource owners nobody can modify the policies).

By observing all these factors, finally, we decide to use XACML to express the access control policies for the users in DOSN. But definitely, XACML is required to extend, or combine with some other method, or language, to support proper authentication of the requesters and enforcement of the access control policies.

4.1.1.6.1 Threats and Suggested Extensions to XACML

Although XACML is widely accepted in the real world as a reference solution for access control, it neither provides any way to support privacy, such as authenticate a user, or verifying the attributes included in access requests, or maintaining the integrity of the policies, nor any technique for enforcing of these defined policies. Therefore, to express the access control policies in OSNs, using XACML is required further consideration regarding authentication, and enforcement. There are some open source web or application servers, such as JBoss Application Server[7], Fedora Server[8], etc., which support XACML authorization polices and has integrated mechanism to enforce these policies.

Moreover, without proper authentication, only by assumption, the requesters provide all the correct information in their requests, increases the possibility of several threats, such as "Identity theft" (malicious users use the attributes of any authorized user during the time of access requests), "Impersonation attack" (using or cloning identity of another authorized user for getting access) etc. In a word, without proper authentication or verifying the requesters' attributes the authorization polices are useless, because unauthorized users may get access to confidential data (private information, pictures etc.), which can be used later for creating *faked profile* for "Sybil attacks" or "profile harvesting", by using gathered data for several purposes beyond the knowledge of the original owner. But for authentication there is a large number of ways, but we need to select a suitable one, which is more applicable to fit with XACML, as well as with online social networks.

4.2 Authentication Support in XACML by SAML

4.2.1 Authentication Types and Factors

Mainly, there are two types [42] of authentication: *User Authentication* to verify a user's claim, and *Entity Authentication* to verify an entity or token, derived during the time of authenticating a user. If we consider, one of the most common scenarios of authentication in a web application, that is, one user can authenticate (user authentication) himself by providing his unique user name and password, or using some other methods; during the time of authentication, the web browser stores a session token, and later uses it with every request for entity authentication. Thus, the user authentication is performed only once for a single session, but the entity authentication is performed during the time of each request.

Depending on the three factors of an authentication method: "Who you are", "What you have" and "what you know", a single authentication mechanism may be combination of all, or any two of them. The multi-factored authentication can be defined [45] as follows:

- **Single-factor authentication** –Only one of the factors is used for authentication, such as providing password (Information revealing).

- **Two-factor authentication** – Any two of the factors are combined within single authentication mechanism, such as to get into a building two things are required: a code (Information revealing) and a key (Something have).

- **Three-factor authentication** – All the three factors are combined together for single authentication, such as to enter into a highly secured place three things are needed- fingerprint against stored fingerprint (Information about who you are), an access card (something have), and secured code (Information revealing).

[7] www.jboss.org/jbossas

[8] http://fedora-commons.org/download/2.1b/userdocs/server/security/AuthorizationXACML.htm

4.2.2 Security Assertion Markup Language

SAML [36, 37, 40] is an XML based language standardizes by the OASIS for exchanging authentication and authorization data among different security domains, such as between an "identity provider" (a producer of assertions) and a "service provider" (a consumer of assertions). SAML assumes every user has enrolled with at least one identity provider. Thus, a service provider relies on the identity provider to identify the subject. At the subject's request, the identity provider passes a SAML assertion to the service provider. Based on this assertion, the requester is authenticated to the service provider.

SAML identified there major problems [37] and trying to solve it:

1. "**Single Sign-on (SSO)**" to simplify identity management across organizational boundaries.

2. "**Distributed transaction**", such that users' can obtain a certificate by giving their credential in one web site, and authenticate themselves by the same in another site, rather than providing the credential again.

3. "**Authorization services**", such that users can access resources based on authorization policies.

4.2.2.1 Structure of SAML

SAML describes XML-based *assertions*, *protocols*, *bindings*, and *profiles*. The overall structure of SAML is illustrated in Figure 4.3.

- *SAML Core* defines the overall structure of SAML *Assertions,* such as syntax and semantics, and the *protocol* defines the rules, how to request, and how to transmit, SAML assertions. *A* SAML *Assertion* is a statement including some features about a subject, such that the service provider can take access decision for the particular request. An assertion includes:

    - *Authentication statement* indicates that a user is authenticated by a specific method, at a certain time.
    - *Attribute statement* defines some attributes and corresponding values of a subject.
    - *Authorization decision* defines the access rights of a subject on specific resource by providing some evidence.
    - The statements may be digitally signed and extend according to the own choices.

All the attached statements in a specific assertion include same metadata [38], such as follows:

- *Assertion ID*: Defines the unique identifier for each assertion.
- *Assertion Issuer*: States, who is the issuer of the assertion.

- *Assertion IssueInstant*: Specifies the time of issuing the assertion.
- *Subject Information*: States the name and security domain of a subject and may include some optional information about the subject, such as public key.
- *Conditions*: Defines various rules for the assertion, such as validation time period, constraint on audience or target, etc.
- *Advice*: Defines how the assertion is prepared.

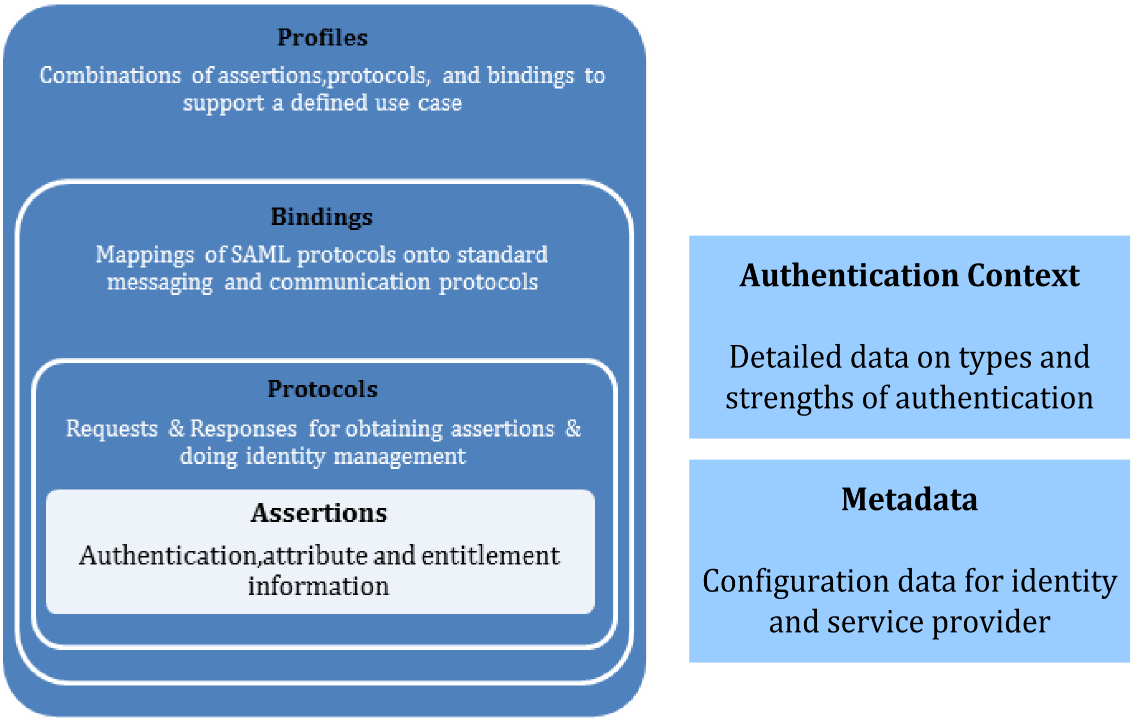

Profiles
Combinations of assertions,protocols, and bindings to support a defined use case
Bindings
Mappings of SAML protocols onto standard messaging and communication protocols
Protocols
Requests & Responses for obtaining assertions & doing identity management
Assertions
Authentication,attribute and entitlement information
Authentication Context
Detailed data on types and strengths of authentication
Metadata
Configuration data for identity and service provider

Figure 4.3. Basic SAML architecture (adapted from [40]).

- *SAML protocol* defines the syntax of the requests and responses, which may be exchanged through different systems. More general, SAML protocol is same as the general http request-response protocol. SAML defines various types of protocols, such as *Authentication Request Protocol, Assertion Query and Request Protocol, Artifact Resolution Protocol, Single Logout Protocol, etc. Query* is the most common and important type of *SAML request* and *assertion* is the most general type of *SAML response*. The relying party (service provider) sends a request to the asserting party (identity provider), including the *query* and the asserting party sends back the response, including *assertion* so that the relying party can take decision about access permission for that user.

There are different types of *request query*:

- *Authentication query* is for requesting authentication information about a specific user.

  - *Attribute query* is for requesting information about certain attributes of a specific user.
  - *Authorization decision query* is for requesting a decision (yes/no) about a specific user to a specific resource, by means of some evidences.

  Sometimes the responses (of the *request query*) include the status information with the assertion(s), or contain only the status information in case of errors. The status information includes *status codes,* such as *Success, VersionMismatch, Receiver,* and *Sender* with optional element *statusMessage* and *statusDetail*. The responses are generally, digitally signed.

- *SAML binding* [39] defines the procedure or method to transform the SAML *requests* and *responses* into basic messages or standard communication protocols. Different kinds of bindings supported by SAML 2.0 are:

  - *SAML SOAP Binding.*
  - *Reverse SOAP (PAOS) Binding.*
  - *HTTP Redirect (GET) Binding.*
  - *HTTP POST Binding.*
  - *HTTP Artifact Binding.*
  - *SAML URI Binding.*

- *SAML profiles* [40] are specific groupings of the *assertions*, *protocols*, and bindings according to the basis of certain application requirements. The profiles supported by SAML 2.0 are:

  - *Web Browser Single Sign-On Profile.*
  - *Enhanced Client and Proxy (ECP) Profile.*
  - *Identity Provider Discovery Profile.*
  - *Single Logout Profile.*
  - *Artifact Resolution Profile.*
  - *Assertion Query/Request Profile.*
  - *SAML Attributes profile:* Define mechanisms for exchanging attributes through the assertions. Different attribute profiles are supported by SAML 2.0, such as *X.500/LDAP Attribute Profile*, *UUID Attribute Profile*, *XACML Attribute Profile*, etc.

4.2.2.2 SAML 2.0 Profile of XACML 2.0

The main drawback of using SAML in online social networks is the expressiveness of authorization policies, which are too limited for presenting fine-grained dynamic access control policies. On the other hand, XACML overcome this problem by providing an enormous set of functions for expressing complex rules on diverse types of resources, but its main drawback, that limits its scope of applicability, is proper authentication of requesters, or verifying attributes attached within requests. Further, XACML does not provide any transport mechanism or protocol to transfer the requests and responses, and does not offer any method to express security constraints to bind the authorities with their authorization policies, which are offered by the SAML *assertion issuer* and *condition* elements.

Therefore, to handle these security issues and to provide privacy-enhanced authentication and access control mechanism, SAML profile of XACML [41] is offered. This profile defines, how to protect, request, and transport the XACML scheme, by combining XACML's authorization policies expressions and evaluation facilities, with the SAML's assertion administration abilities.

In Figure 4.4 communications among all the standard components in SAML profile of XACML are presented, but it is not mandatory to include all the components for every application. *Attribute Authority* (AA) is responsible for issuing *Attribute Assertions,* which presents the attributes of subjects. Three types of attributes are used in this profile: *XACML Attributes* for *XACML Request,* are represented by the element *xacml-context: Attribute* element; *SAML Attributes* for *SAML Assertion,* are represented by the element *saml:Attribute,* or may be related to a subject to present the attributes by the *saml:SubjectStatement* element; *XML attributes* to present the properties of a subject, or object, or resource, or environment. *Policy administration point (PAP)* issues the authorization policies, which are expressed as the SAML *policy assertions*. The components *Policy decision point (PDP)*, *policy Enforcement point (PEP),* and the term *policy*, have same meaning and purpose as XACML.

This profile defines mainly, four specifications:

1. **How to use SAML attributes in an XACML system**: In order to use SAML *AttributeAssertion*, *AttributeStatement,* and *Attributes,* for transmitting or storing attributes and to use the SAML *AttributeQuery protocol* to ask for attributes within an XACML *Request Context*, the SAML attributes need to map into the XACML attributes. Each *saml:Attribute* element in a SAML *AttributeAssertion* is mapped into single *xacml-context:Attribute* and for attribute values the corresponding *saml:AttributeValue* is used in the *xacml:Attribute-Value* and the issuer of the SAML *Assertion* is the issuer of all the *xacml-context:Attribute* elements.

2. **How to use SAML for request, response, store, or transmit Authorization decisions in XACML system:** SAML defines *AuthzDecisionQuery* protocol and *AuthzDecison-Staement* assertion to handle the requests and corresponding responses for handling authorization decisions. But the main problem is that they are not enough to carry all information, which the XACML requests and responses contain. With the objective of supporting the XACML *Request* and *Response Context* with the SAML *Request* and *Response* two extensions are provided: *xacml-samlp:XACMLAuthzDecisionQuery,* offers the *PEP* to submit an *XACML* request within *SAML* request with other optional information; and *xacml-saml:XACMLAuthzDecisionStatement,* offers the XACML *PDP* to return XACML *Response Context* as a response of the *xacml-samlp:XACMLAuthz DecisionQuery* to the *PEP,* where the XACML response may be stored or conveyed as the SAML assertion.

3. **How to use SAML for request, response, store, or transmit XACML policies:** SAML does not offer any facility to transport, or store authorization policies, but XACML states two schema elements *policy* and *policySet* for authorization policies. That's why, to support request, response, store, and transmit of the XACML *Policy* or *PolicySet* as part of

the SAML requests and responses, this profile offers two new elements. The element *XACMLPolicyQuery* is to extend the functionality of the PDP by supporting, request for XACML Policy or policySet from the PAP as a part of the SAML Request. And *XACMLPolicyStatement* is either used in the SAML Response provided by the PAP as the response of the *XACMLPolicyQuery,* or the SAML *Assertion* for storing in a repository.

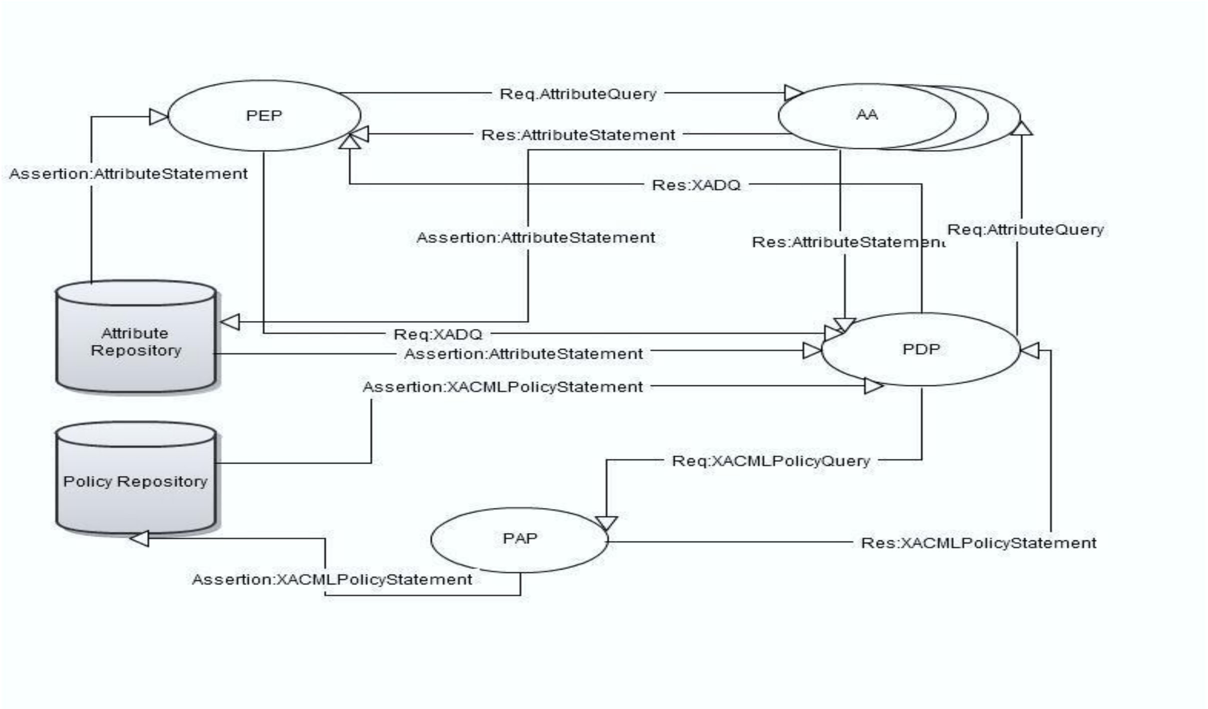


Figure 4.4. SAML profile of XACML communication diagram (adapted from [41])

4. **How to use SAML Assertion, Request, and Response:** *SAML Assertion* is used to encapsulate the authorization decisions, or policies, or attributes, which may be signed by the issuer, and the attributes are mapped into XACML attributes. SAML *Assertion* may contain an additional optional element named, *SubjectConfrimation* to present the conditions to use that Assertion by a relying party; or optional element termed *conditions,* which are considered as XML attributes by the *PDP*, before evaluating the policies contained on that assertion. *SAML Request* is used to bind a query for authorization decision, or policy, or attribute, and may be signed by the requester. *SAML Response* is used to return response of the queries, and contains the authorization decision statement, or policy statement, or attribute statement, which may be signed. It also includes the element *Status* (depend on the XACML *status* element), to denote additional information, such as syntax error, missing attributes, etc.

In a whole, this profile defines six types of extensions [41] of standard SAML:

- *AttributeQuery*: Standard SAML *Request*, which is used to query about subjects' attributes from the *Attribute Authority*.

- *AttributeStatement:* Standard SAML *Statement*, which is used, either in the SAML *Response* from the *Attribute Authority*, or in the SAML *Assertion* from an *Attribute Repository*.

- *XACMLPolicyQuery:* Extension of the SAML *Request,* which is used to query about XACML policies form the *PAP*.

- *XACMLPolicyStatement:* Extension of the SAML *Statement*, which is used either in a SAML Response from the *PAP*, or the SAML Assertion from a *Policy Repository*.

- *XACMLAuthzDecisionQuery:* Extension of the SAML *Request*, which is used by the *PEP* to query about the authorization decisions to the XACML *PDP*.

- *XACMLAuthzDecisionStatement:* Extension of the SAML *Statement*, which is send back within SAML *Response* from the XACML *PDP* to the *PEP*.

4.2.2.3 SAML Supported Authentication Methods

Before selecting an authentication method for the users in OSNs, we need to discuss about various types of authentication methods supported by the SAML [43]:

- *InternetProtocolPassword*: Authentication based on the subject's IP address along with user-name and password.

- *Kerberos*: Local authentication of a user by his password, in order to obtain the Kerberos ticket for further authentication in the network. Pre-authentication data type and other metadata related to the Kerberos ticket is supplied to the local authentication authority.

- *MobileOneFactorUnregistered*: Authenticates mobile devices rather than the user, without user interaction. It is an important procedure in an environment, where the device authentication is necessary for the security purposes.

- *MobileTwoFactorUnregistered*: Two-factor authentication method for both the device and user, such as the mobile device and the user's secret key.

- *Password*: Authenticate a user based on his provided password to the authentication authority over an insecure HTTP session.

- *PasswordProtectedTransport*: Authenticate a user based on this provided password to the authentication authority over a protected session. (Transport Protocol Type may be SSL, IPSec, Mobile Network Radio Encryption, etc.)

- *Public Key - X.509*: Authenticate a user by his digital signature, where the key was certified previously, by an X.509 public key infrastructure.

- *Public Key - PGP*: Authenticate a user by his digital signature, where the key was certified before, by a PGP public key infrastructure.

- *Public Key - XML Digital Signature*: Authenticate a user by his digital signature, according to the rules, presented in the XML Digital Signature Description.

- *Smartcard*: Authenticate a user to the authentication authority by a smartcard, such as cryptographic challenge/Response, cryptographic calculator (onetime password), etc.

- *SmartCardPKI*: Authenticate a user by a two-factor authentication method, where a smartcard (contains a secret key) is used with a PIN.

- *SoftwarePKI*: Authenticate a user by X.509 certificate, where the certificate is kept in software.

- *Telephony*: Authenticate a user by his personal fixed-line telephone number, where a telephony protocol, such as ASDL is used for the transportation.

- *Telephony ("Nomadic")*: Authenticate a user during roaming by his line number, suffix along with his password.

- *Secure Remote Password*: Authenticate a user by Secure Remote Password (SRP) [44]. The SRP is a password-authenticated key agreement protocol, which ensures protection against passive and active attackers.

- *SSL/TLS Certificate-Based Client Authentication*: Authenticate a user by means of a client certificate, where the transportation protection is ensured by the SSL/TLS.

4.3 Key Factors to select an authentication method in OSN

After considering the SAML profile of XACML, the main question is, which authentication method is suitable to use in the perspective of online social networks. Therefore, before selecting a specific authentication method to verify requestors' claims for accessing a resource in distributed online social networks, we point out several issues for careful consideration:

- The primary and most important factor, the method should maintain balance between performance and usability. As the subscribers or users in online social networks are from different sectors, and it is common that most of them are not properly aware about security and privacy over their data. Therefore, if the method is secured enough but cumbersome for the users, such as they need to maintain a large amount of information for each access request, or its takes large numbers of steps for authentication, then definitely, uncertainty arises about its acceptability. On the other hand, if the method is easy to use for all the users, but too weak that attackers can easily break it and get access to the private resources. And afterwards, use this information beyond knowledge of the original owner, which may be social embarrassment for that user, then again, some sort of uncertainty arises about its applicability.

- Another concerning factor, the method needs to be supported by SAML, or may be extended to support by the SAML, or may be a complete new framework but necessarily, fit with the XACML for the authorization control.

- Further, it is necessary to consider which type of authentication method is the most suitable for OSNs, such as password or secret based authentication, or HTTP based authentication, or form based web authentication, or certificate based authentication, or credential based authentication, or token or key based authentication.

- Finally, some other issues need to take into concern because of the decentralized structure, such as secure storage of the identity information, or how much information a requester should disclose during the time of request.

From the existing available authentication methods, it can be summarized, to authenticate a user, he must need to submit some credentials so that the attributes, or information that is attached in the credentials can be verified to authenticate that user. Some most common types of credentials used by the users are: password, secret key, biometric, certificates, and security tokens. The biometric based authentication is out of scope, because in online social networks, subscribers are from different sectors, and they try to use it from different types of machines with diverse platforms. Moreover, the biometric based authentications, such as retina, or heartbeat, or fingerprint scanning, etc. are costly and somehow, depend on underlying machines and platforms.

The most commonly used password based authentication is not logical to authenticate a user, or verifying his attributes during the time of access request in OSNs. In DOSN, the profiles and private resources may be stored on the owner's own machine, or in some other machines (replicas). Therefore, when a user sends an access request to the owner, or replicas, it is more logical that the requester should submit a secret key for authentication, which is previously set by the owner, rather than the requester. Although the secret key based authentication seems very promising for OSNs, many factors require careful consideration to protect against different threats. To ensure the quality of the secret key, it is a primary task to make it unpredictable for other users. As in OSNs, the social connections list reflects real life relationships. Therefore, it is quite common to use some common information (between the requester and owner) as secret key, such as high school mathematics teacher name, first pet animal name, first meeting place, etc. However, it is also necessary to make the key unpredictable to other users, who know this information but don't have same access privileges. Thus, we suggest to exchange, and determine the secret key between the owner and his social friends in out-of-band (face to face meeting, by telephone, etc.) such that it is hard to guess by their mutual friends. Further, a pseudonym may be included in the request as a requester identification, which is also provided by the owner, but the pseudonym needs to choose on the basis of social relationship. Obviously, it gives extra protection but needs to address many issues to make it applicable. Therefore, we excluded this from our current focus, but consider as a great possibility for the future work.

Finally, the certificate based credential is becoming a popular way of authentication among different areas because people inspire from the real world scenarios. As for example, if we consider the scenario of authentication in the computer rooms at computer science department in

KTH, one student needs “Photo ID” to proof his identity, “Student Card” to proof as a student of KTH, and “Access Card” to proof as a student of CSC. Each of these proofs can be considered as a certificate containing some credentials, and they are related but not dependent on each other. Digital certificates, such as X.509 certificates, digital identity cards, LDAP, Kerberos tickets etc. are commonly used in the web systems for authentication, as well as for exchanging individual public keys of the users. It is better than password based authentication, as in this case the user does not need to remember, or store large numbers of user-name and password combinations, but enjoy enhanced privacy. Large numbers of works are going on to investigate the applicability of this mechanism and make it usable through XACML, or through the combination of XACML and SAML.

4.3.1 Related Solution and Unanswered Questions

Claudio A. Ardagna et al. [16] propose an extension of XACML and SAML, and then combine both to support privacy-enhanced credential-based access control for the users. The authors assume that the enforcement point has complete knowledge about all the attributes of all the requesters. The extension provides the requesters, an opportunity to get informed about which specific attributes of a credential they need to disclose to access a resource. A credential is a combination of some attribute names and corresponding values, and also includes the attributes types, and the issuer as metadata. The extended architecture is presented in the Figure 4.5. The proposed architecture works in two rounds. In the first round the requesters specify the resource they want to access, and acquire the relevant policies (XACMLPolicy assertion element). Then in the second round, the requesters send the same request again, but this time with their claim (in SAML) by submitting evidence (attribute values, or conditions over attributes without disclosing actual values) to get access permission. In order to support this architecture, XACML is extended to express credential-based access control policies. This extension offers an advantage for the policy writers to present some conditions, or actions in the credential, which must be fulfilled or performed by the requesters to get access permit. Also the SAML is extended to support conditions on the attributes values as well as statements about the states of the provisional actions.

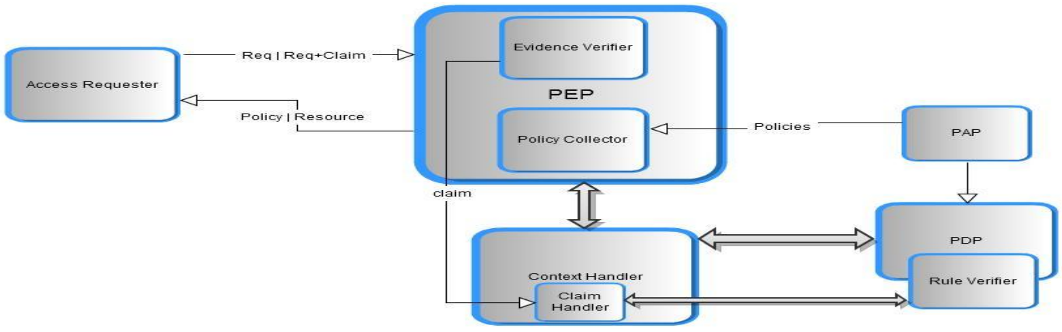


Figure 4.5. Extended architecture of XACML (adapted from [16]).

The proposed architecture enhances the privacy of users' information as the authors suggest using anonymous credentials for the requesters. But in the first step of the proposal, where a requester submits his request only for gaining knowledge about related policies for a specific resource, increases the chance of different types of attacks. Because in this case, the adversaries can easily have knowledge about the policies and thus, use this knowledge afterwards to get unauthorized access. Moreover, the assumption, "enforcement point knows all the requesters attributes", needs proper concern to make this proposal applicable in DOSN.

However, before using digital certificates in online social networks, we need to find out answers of some questions:

- Without any central authority, if the owner issues certificates for all his social links, then how many certificates each user maintains to access individual user's information.

- Are all the users ready to store all the certificates on their machines, or does it become a burden on the users, which drop its acceptability?

- How will the certificate of the users be managed (such as generating user id or unique id for the certificate of each user)?

- How will the revocation of the certificates be managed?

Chapter 5

Implementation of Access Control Policies

This chapter presents implementations of some common access control policies for OSN users, in Sun's XACML, one of the open source implementations of OASIS standard XACML version 2.0. The access control policies are depicted through the use of general explanation of some real world scenarios.

5.1 Open Source Implementations and Why Sun XACML?

Among different existing implementations of XACML, some of the implementations are open source and some of them are proprietary. Again among all the open source implementations, some offer only standard features of XACML (specific versions), such as supported functions, data types, algorithms, etc. where some other provide some extended features, such as *indexing*, for efficient searching of policies; *attribute finder*, for locating missing attributes from storage, etc. Several currently available open source APIs of XACML are: Sun's XACML[9], XACM-Light[10], XACML Enterprise[11], PicketBox XACML[12], Authorization API[13], etc.

Our main plan is to shed some lights about the possibility of presenting access control policies using the existing features of XACML 2.0, and try to find out its possible extensions in the perspective of OSN. XACML access control policies have quite complex structure, as in most of the cases; each policy is combined by different rules. Therefore, there is a possibility of conflicts of these rules with each other, where it is compulsory to evaluate all the rules perfectly to get correct decision. For this reason we search for an open source API of XACML, which is considered as low level API and closely similar to standard XACML, regarding to the policy syntax and semantics. In this condition, sun's XACML is the best choice. Moreover, sun's XACML is one of the APIs, which is considered as foundation of many other applications and, its large acceptance in both commercial applications and also in research assignments indicates it's free from bugs. Further, this API supports the full features of XACML 2.0, as well as it can be extended by adding new features, such as functions, or data types, or algorithms, etc. Furthermore, this API has the facility to parse access control policies, and requests or responses, and includes a sample implementation of *PDP* (policy decision point) to evaluate policies from both local machines and by URL. Finally, the API is written in the java programing languages and only requires java 2.0 software to run the tests. Therefore, based on all of these above considerations, we choose sun's XACML to test the policies/requests use cases.

[9] http://sourceforge.net/projects/sunxacml/
[10] http://sourceforge.net/projects/ xacmllight/
[11] http://code.google.com/p/enterprise-java-xacml/
[12] http://www.jboss.org/picketbox/downloads.html
[13] http://www. oasis-open.org/committees/document.php?document_id=33416

5.2 Experiment Environment

The test environment for evaluation of the access control policies is set up as follows: a laptop with windows vista operating system, 2.7 GHz AMD Sempron processor, 2GB Ram, 250 GB hard drive, and NetBeans IDE 6.7.1[14] is installed in the machine.

5.3 Major Issues for Providing Access Control Policies in OSN

According to the requirements of access control policies for a user in OSNs, and initial architecture presented in the chapter 3, we can summarize the subjects, resources, conditions, and access types in OSN as follows: (depicted in table 1)

Subject	Resource	Access Type	Factors for Rules/conditions
A user in OSN.	**Profile** of a user in OSN. • **Resources,** such as wall, pictures, etc. • **Relations,** such as friends, family, etc.	**View/Read** **Write/Comment** **Tag**	**Subject** • Static properties, such as relationship type. • Dynamic properties, such as age. • Group or team of users, such as divide users based on an attribute “trust”. • Assigned roles, such as divide members in a group based on roles. **Resource** • Properties, such as title. • Group of resources, such as four different pictures albums under the resource “picture”. **Context** • Time, such as access time limited by working hours. • Date, such as access on an event schedule is bounded by a specific period. **Obligations** • Additional constrains, such as require email address before comment on a document. **Empty condition** • No specific rule, such as basic information of a user is available for every profile searcher in an OSN.

Table 5.1. A summary of access control factors in an OSN

[14] http://netbeans.org/

5.3.1 Assumption

All the access control policies have presented in this chapter are designed on the basis of standard XACML 2.0 (syntax and semantics). Corresponding policies are stored on users' personal machines, and they have their own web server (supports XACML authorization policies) to enforce these access control policies. Our main focus is on policy expression on the basis of diverse conditions. We propose to store social relationships information of users' with individual unique identification, such as email id or user name, and corresponding attributes, such as trust level, relationship, etc. Therefore, when a user sends an access request to the owner for his profile, the necessary information to evaluate corresponding policies are obtained, either from the requester's request, or from the repository of stored attributes. Finally, enforcement of access decisions is completely depended on the web server.

5.3.2 Scenarios

We present a variety of use cases (scenarios) from OSN to depict several issues, for authorization control to users' profiles. The corresponding XACML policies are presented in **Appendix A**.

Scenario1: Rahim (*rahim@gmail.com*) and Karim (*karim@yahoo.com*), two subscribers registered in an online social network (OSN) by their email addresses, which are considered as unique. In OSN both of them have their own profile, which contains their private information, uploaded resources, such as pictures, videos, etc., other types of resources, such as wall, group-pages, etc., and their social connections. Assume that Karim is socially connected with Rahim. Recently, Rahim went to a party and uploaded some pictures of that party in his profile as an album named "After-Exam-Party". He would like to share this Album with Karim and write an access control policy so that only Karim is able to view the album. All other access requests from his social connections are denied.

Success environment:

- All the pictures from the album "After Exam party" are only accessible by Karim.
- Karim can only watch (*view* permission) the pictures of the album.

Failure environment:

- Any other person, who is already connected with Rahim in the OSN, is able to access any picture from the album.
- Karim is able to do any other operation except view such as delete, comment, etc. on the pictures.

Access Type	Condition to get access permission	Policy Description
View	Requester's id= karim@yahoo.com	Policy based on requester's id in OSN.

Table 5.2. Access control factors for scenario1

<u>Scenario1-Extension1</u>: The above scenario is extended such that Karim can access the album "After-Exam-Party" by providing any one of his two email addresses: *karim@yahoo.com* or *karim_bd@gmail.com,* where both addresses are used as identifier of Karim in OSN.

Success environment:

- Karim gets access permission for the album by providing any one of his two email addresses.

Failure environment:

- Access request of Karim is denied, if he requests using the email address *karim_bd@gmail.com*.

Access Type	Condition to get access permission	Policy Description
View	Requester's id= karim@yahoo.com *OR* karim_bd@gmail.com	Policy based on multiple IDs of requesters.

Table 5.3. Access control factors for scenario1-extension1

<u>Scenario1-Extension2</u>: The scenario.1 is extended such that now, Karim has more access privileges than before. Until now, he is only able to view the pictures, but now, he can comment and tag himself on any pictures of the album.

Success environment:

- Karim is able to view, or comment, or tag himself on any pictures form the album "After Exam Party".

Failure environment:

- Karim is unable to comment or tag himself on any picture.

Access Type	Condition to get access permission	Policy Description
View Comment Tag	Requester's id= karim@yahoo.com *OR* karim_bd@gmail.com	Policy based on requester's multiple IDs, and single policy for different types of operations.

Table 5.4. Access control factors for scenario1-extension2

<u>Scenario2</u>: Rahim and Karim, two subscribers in an OSN, store their social connections list with an attribute "trust level" and its corresponding value, which is "High", or "Medium", or "Low".

Recently, Rahim has celebrated his birthday with some of his close friends. At the birthday party, he has recorded a number of videos. Among these all videos, he uploads a private video titled, "Birthday_Celebration" in his profile, and wants to share it with highly trusted (trust level is "High") from his social connections list. Rahim set the attribute "trust level" to "High" for Karim, but this attribute is not disclosed to him. As the attribute "trust level" is highly confidential and has large impact on social relationship, therefore, the attribute and its value is stored in a secured repository and not disclosed to anyone except the attribute setter (individual user). Therefore, when a request comes to access the video "Birthday Celebration" the conforming value of the attribute "trust level" will be obtained from the repository, and depends on that access permission is accepted or rejected.

Success environment:

- Karim is able to access the video "Birthday Celebration" but not aware about value of the attribute "trust level" and that's why, does not include this within his access request.

Failure environment:

- Karim has knowledge about the attribute "trust level".
- Some other from Rahim's social connections list, but not highly trusted (trust level is "Medium", or "Low") gets access permission for the video.

Access Type	Condition to get access permission	Policy Description
View	Requester's trust level = High	Policy based on requester's attribute "trust level", which is defined by the owner.

Table 5.5. Access control factors for scenario2

<u>Scenario3</u>: Rahim and Karim, two subscribers in an OSN, decide to maintain their social relationships list with an attribute, named "relations" to indicate the relationship between subscribers and their social links, such as friends, colleagues, family, relatives, etc. Recently, Rahim uploads a document in his profile, titled "Weekend parties", and desires to share it with people, who have friend relationship with him in OSN. Therefore, Rahim defines one policy so that all friends from his social connections list can access the document. Karim is marked as friend in Rahim's social connections list. Therefore, when Karim directs his access request with his email address (unique for each user in OSN), his relationship with Rahim is obtained from the repository, and according to the basis of this relationship the access permission is granted.

Success environment:

- Karim can access the document "weekend parties", but does not need to include "relation" attribute within his access request.

Failure environment:

- Without including “relation” attribute, Karim cannot access the document.
- Someone from Rahim’s social links, but not treated as friend gets access permission.

Access Type	Condition to get access permission	Policy Description
View	Requester’s relationship with owner = “friends”	Policy based on requester’s relationship indicator attribute “relations”, with the owner.

Table 5.6. Access control factors for scenario3

<u>Scenario4</u>: Rahim and Karim, two subscribers in an OSN, recently, store their social connections with two different attributes, called “trust level” and “relations” with their corresponding values (same as the previous two scenarios). Rahim is working in a company as a marketing director, and needs to submit a secret bidding quotation within few days to acquire a highly profitable contract. That’s why, he makes and uploads a document, named “bidding quote” in his profile and wants to share with some extremely trusted colleagues from the company (that means, trust level =”High” and relations=”colleagues”) for getting valuable feedback. Karim has both of these properties and therefore, when he forwards his request, he gets permission to view and comment to the document.

Success environment:

- Karim is able to access the document “bidding quote”, but not aware about the attributes, “relations” and “trust level”. Therefore, he does not include these attribute inside his access request.

Failure environment:

- Karim has knowledge about the attributes, “relations” or “trust level”, which was defined and stored by Rahim.
- Some other from Rahim’s colleagues list but not highly trusted by Rahim, gets permission to access the document. Again, somebody, who is highly trusted by Rahim but not in his colleagues list gets access permit.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester’s relationship with owner= “colleagues” AND trust level = “High”	Policy based on requester’s multiple attributes, “relations” and “trust level”.

Table 5.7. Access control factors for scenario4

Scenario5: The scenario is same as above, that is, all the users in an OSN store their social connections with two attributes, “relations,” and “trust level”. Currently, Rahim has uploaded one picture album titled “study abroad”, which contains the pictures during the period of his study in abroad. He wants to share this album with his friends list as well as with his family members (not the common people rather than all the people from both lists). Therefore, Rahim defines an access control policy for the album “Study abroad” so that every member from his friends and family list can view and comment on the pictures as well as tag themselves.

Success environment:

- Anybody, who has friend or family relationship with Rahim, is able to access the album.

Failure environment:

- The album is only accessible either by friends, or family members.
- Any person, who has other types of relationship with Rahim, such as colleague, relative, etc. gets access permit for the album.

Access Type	Condition to get access permission	Policy Description
View Comment Tag	Requester’s relationship with owner = “friends” OR “family”.	Single policy for multiple groups, depends on the attribute, “relations”.

Table 5.8. Access control factors for scenario5

Scenario6: To celebrate the upcoming national day of Rahim’s home country, he decides to arrange a programme after seven days from today. He has made an event, which contains his initial thought about the programme, that is, schedule, venue, and details description of the each and every sub event. But to make it more worthful he wants some suggestions from all of his connections in the social network. Therefore, he uploads the event titled “National day Event” with detailed description and makes it available for all socially connected people such that they can read and comment for next five days from today. In order to confirm this he writes a policy so that the event is accessible only for next five days.

Success environment:

- Anyone from the social connections list of Rahim is able to read and comment to the event for next five days from today.
- From the sixth day, nobody can access the event.

Failure environment:

- Anybody form Rahim’s social connection can access the event after five days.
- Anybody, who is not from existing social connections list of Rahim, can access the event.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester's request date is valid for next five days.	Policy based on requester's request date.

Table 5.9. Access control factors for scenario6

Scenario7: Rahim starts a new group page about "security requirements in online social network", and wants to share with all from a particular social network service to view and put their comments. The group page is for exchanging thoughts of general people, who don't have in-depth knowledge about any technical means in security, and also for the people, who have rich knowledge about security issues. But Rahim is aware about one thing that some wrong comments may create wrong conception among general people and he thinks that the probability of giving unaware comments to the document is high after working hours (from 6.01 am to 5.59 pm). Therefore, he creates a policy such that nobody can comment to the group page from 6pm to 6am.

Success environment:

- Anybody, who is invited by Rahim, is able to read and comment to the group-page from 6.01am to 5.59pm.

Failure environment:

- A person is able to make comment in the group page after 6pm or before 6am.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester's request time > 6.00 am *AND < 6.00 pm*	Policy based on requester's request time.

Table 5.10. Access control factors for scenario7

Scenario8: Rahim is working in a big company, which is constructed as combination of some disjoint buildings located very close to each other. Rahim discovers, sometimes it is difficult and time consuming to meet with colleagues physically, or to discuss in telephone for long time, specially, with employees from different departments. Therefore, he creates a new group page named "Information Zone" for discussion about official works. Thus, he defines a policy so that the colleagues from his social connections list can comment to the discussions on the group-page during the working hours within working days. (Monday to Friday within 8am to 5 pm)

Success environment:

- Any colleague of Rahim can access the group page and put their opinion within working hours (8am to 5 pm) in any working day (Monday to Friday).

Failure environment:

- A person gets permission to comment in the group page on weekends.
- Somebody is able to access (comment) to the group page after or before working (Rahim's company) hours, that is, from 5.01pm to 7.59am.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester's relationship with owner=colleagues AND 8.00am ≤ request time ≥ 5.00 pm AND request day ≠ Saturday or Sunday	Policy based on multiple rules, such as relationship with owner, request time, and request day.

Table 5.11. Access control factors for scenario8

<u>Scenario9:</u> Recently, Rahim watches a horror video named "supernatural activity", which contains some clips that may distress young people. But Rahim thinks, it is an interesting video, and thus, wants to share it with people from his social connections list. But he defines a constraint that people, who are more than twenty-two years old, are able to watch it. Hence, he states a policy so that anybody from his existing social connection in OSN, who is more than or just twenty-two years old (age is calculated during the date of request), can watch the video.

Success environment:

- Any person form the social connections list of Rahim, who is equal or more than twenty-two years, is able to watch the video "supernatural activity".

Failure environment:

- Anybody, who is socially connected with Rahim, but age, is less than twenty-two years old gets watch permission.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester's request date – date of birth ≥ 22 years	Policy based on requester's age, that is, calculated from his birth date and request date.

Table 5.12. Access control factors for scenario9

<u>Scenario10:</u> Few days back, Rahim has moved out for hunting in a wild forest in Brazil. In this tour he took lots of photographs as well as captured some videos about wild animals' behavior, hunting experiences, interview of the local people, etc. But he is not interested to disclose any information about this tour to anyone. Rahim has been a member of a hunting club, named "Hunting Techniques" for last four years. Yesterday there was a meeting among the club members and Rahim found that two of his social friends (Karim and Jishan) joined to the club recently. They are very much interested to have some knowledge about hunting in a wild forest. Therefore, now, Rahim decides to share one video, "Nature of Wild Animal" and one picture album, "Life in a Forest" from that tour. Thus, he defines one access control policy for both of these resources (different types), and gives access permit to both of his friends.

Assumption:

Assumption is two friends of Rahim who recently joined to the hunting club have different relationships with Rahim –one is his friend and another one is his colleague. Moreover, they don't have any predefined properties (such as age, trust level, etc.) in common. Further, all the members of the hunting club have unique user_id issued by the club authority.

Success environment:

- Karim and Jishan are able to watch the video and pictures, by providing their user identification and issuer of the id (member id provided by the club authority).
- Rahim offers single policy to control access to the both resources.

Failure environment:

- Any person from Rahim's social relations list except Karim and Jishan granted access permission.
- Rahim needs to define multiple access control policies, for these two different types of resources.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester's member ID in "Hunting Techniques" (a hunting club) = karim Hasnat OR Jishan Hasib	Policy based on requester's member id (not that one used in OSN), and single policy to control access to different types of resources.

Table 5.13. Access control factors for scenario10

Scenario11: Couple of months ago, Rahim came to Sweden for doing his masters in KTH. At the beginning, he faced some challenges, such as accommodation arrangement, shopping, educational system, transport system, etc. Now, with the time he gains knowledge, about how to handle these issues. But he wishes to share this experience with people, who are planning to come in Sweden for study in near future, before their arrival. Therefore, he composes a document, which contains lots of information about these common issues. The document is uploaded, and titled as "Before coming to Sweden" to indicate the content of the resource (document).

Success environment:

- Any person, requests by including the keyword "Sweden" as document id is able to access the resource “Before coming to Sweden”.
- Anybody sends his request by adding any other words, before or after the keyword “Sweden” is also able to access the document.

Failure environment:

- Without including the keyword “Sweden” within access request, somebody gets access permit.

Access Type	Condition to get access permission	Policy Description
View	Request contains the keyword “Sweden” as resource ID.	Policy based on keyword, which is part of the resource id.

Table 5.14. Access control factors for scenario11

<u>Scenario11-Extension1</u>: Recently, Rahim has followed; some rules have changed in Sweden and will be applicable to international students from the year 2011. Therefore, he needs to change the document “Before coming to Sweden”, but he plans to make a new version of the document to include all the recent changes. Moreover, he keeps the old version without doing any modification. Therefore, he just keeps the same title for both versions of the document, but adds one new property “version information” (current and old) to differentiate between these two forms.

Access Type	Condition to get access permission	Policy Description
View	Request contains the keyword “Sweden” as resource ID AND version information = current or old.	Policy based on keyword, which is part of resource id, and version information to identify the latest one.

Table 5.15. Access control factors for scenario11-extension1

Success environment:

- Any person sends access request by including the keyword "Sweden" as resource id, and "current" or "old" as version information, gets access permit to the current or old document (According to the version information).

Failure environment:

- Without providing the keyword “Sweden” somebody gets access permit.

- Without providing version information someone gets access to any version of the document.
- Someone requests for one version of the document and gets access to another version.

Scenario12: Rahim has uploaded a large number of picture albums under pictures (one of the resource element) and each of the picture album is identified by the album name, such as “After Exam party”, ”Study Abroad”, “Life in a Forest”, etc. These albums are stored as hierarchical structure, such as “Picture” is the root element then all other albums (mentioned above) are stored under this element, and again, some other albums, such as “After-first-sem”,”After-second-sem”, are uploaded under the album “After Exam Party”. Rahim wants to share the whole “Picture” resource (all the albums) with his those social friends, whom he trusts most (trust level high) and also from his friends list. Therefore, Rahim express one policy to define access rights for all the albums.

Success environment:

- A person form Rahim’s friends list and have high trust level, can access all the albums by sending one single request.
- A person form Rahim’s friends list and have high trust level is able to view and give comment on the pictures.

Failure environment:

- A person, who is not in the Rahim’s friends list, or don’t have high trust level, can access any picture from any album.
- An authorized person needs to send multiple access requests for different albums.

Access Type	Condition to get access permission	Policy Description
View Comment	Requester’s relationship with owner=friends AND trust level = High.	Policy based on hierarchically stored resources, that is, one policy to control access to large number of resources.

Table 5.16. Access control factors for scenario12

Scenario13: Rahim is a master’s student of information security in a university. He is very much interested about the security issues in “peer-to-peer systems”, that is, security requirements, current architectures and comparative drawbacks, existing threats, etc. in a p2p system. Thus, he makes a group “P2P Security” and invites people, who are interested in this same area, and also who have deep knowledge about this issue. Rahim made the group as a closed group so that everybody cannot join to the group directly by sending a single join request. To be a member of the group, “P2P Security”, a user must be invited by some other group members. Therefore, to run the group in a professional and charming way, Rahim thinks to divide the members of the group and assign them specific roles, and basis on these roles the access control policies are defined. Rahim create three different roles: “Administrators”, who give permission to users to join to the group, and able to read, modify, and comments on discussions; “Professional Members”, who can read, modify, and comment on any topic; “Ordinary Members”, who can only read the discussions. Each member of the group is assigned a particular role (assigned by Rahim), and this role determines the access rights for specific users.

Success environment:

- A person is able to perform operations in the group, “P2P Security”, on the basis of his assigned role.

Failure environment:

- A person assigned one role but able to perform actions, which is defined for another role.

Access Types based on roles		Condition to get access permission	Policy Description
Administrators	Join Discard Read Modify Comment	Requester’s role = Administrators OR Professional Members OR Ordinary Members.	Policy based on roles and the roles are assigned to each members of the group, statically.
Professional Members	Read Modify Comment		
Ordinary Members	Read		

Table 5.17. Access control factors for scenario13

Scenario13-Extension1: Same as the above scenario, but now a new dynamic condition is defined for the “professional Members” in the group. Currently, a new attribute “date of join”, is added to keep track, when a user become member of the group. Every ordinary member enjoys immediately, all the access rights as “Professional Members”, when they pass two years as a usual member of the group. In this situation, when a member sends his request as an ordinary member, his joining date to the group is compared with the current date and based on this compared values, he will enjoy the appropriate access rights either as “Professional Members”, or as “Ordinary Members”.

Success environment:

- A person is able to perform operations in the group “P2P Security”, on the basis of his role and the date, when he joined to the group.
- An ordinary member, who already passed two years in the group, is able to perform operations as a “Professional Members” of the group.

Failure environment:

- A member, who already passed two years as an “Ordinary Member”, but has same access rights as before.
- Need to change the role of individual member from “Ordinary Members” to “Professional Members” manually, to ensure appropriate access rights.

Access Types based on roles		Condition to get access permission	Policy Description
Administrators	Join Discard Read Modify Comment	Requester’s role = Administrators OR Professional Members OR Ordinary Members AND date of join> 2 years.	Policy based on roles, where each members of the group assigned a role initially, and the roles are changing dynamically based on an attribute.
Professional Members	Read Modify Comment		
Ordinary Members	Read		

Table 5.18. Access control factors for scenario13-extension1

Scenario-14: Rahim is registered in an OSN, to maintain communication and to share approaches and experiences, with his all social contacts, such as friends, relatives, colleagues, classmates, etc. The main benefit of using social network is to keep in touch with each other, and also has up-to-date information about their current activities. Rahim observes that it is quite necessary to maintain secure management of his information (information related to the profile of a user). But he also figures out the importance of providing some basic information to the people, who search for his profile. He follows that some old friends with whom he doesn’t have communication for long time, search for his profile, and also some people do the search randomly. Therefore, to make the profile more secure, when anybody searches for his profile, Rahim offers one policy so that the basic information is available for all, but a notification mail (include, requester’s id and access time) is sent to him (rahim@gmail.com) after every access.

Success environment:

- Any user of the social network is able to view some basic information about Rahim. But when he searches for Rahim’s profile, a notification email is sent to Rahim’s email address, containing identification of the requester and access time.

Failure environment:

- Any user gets permit to view basic information about Rahim but Rahim does not receive any notification email.
- More than basic information (set by Rahim) about Rahim is available to every information searcher.

Access Type	Obligation after access permission granted	Policy Description
View	Notification email = requester's ID and Accessed time, sent to Owner's email address.	General policy without any constrains, but includes an obligation that must be carried after each access.

Table 5.19. Access control factors for scenario14

For all the above use cases, we primarily, aim to describe a number of common scenarios to illustrate a wide range of constraints for access control to the users' profiles in DOSN. During the time of describing use case scenarios, we have considered the requirements, presented in the chapter 3. In the previous chapters, we depicted the idea of using rule-based access control in OSN and presented the rationales behind the selection of XACML for expressing access control policies. After implementing, all these above access control policies in XACML we strict to our design choices. However, it is not possible to describe all the diverse access control requirements of users in DOSN, there is a scope to include new requirements as use cases and express in XACML.

Chapter 6

Analysis

This chapter starts with our proposed approach and its scope of applicability in distributed online social networks. The subsequent sections illustrate the rationales behind our design choices and describe attacker model. The remainder of the chapter deals with a number of alternative approaches that can be applied in DOSN to enhance privacy of the users.

6.1 Proposed Approach

The primary focus of our proposed mechanism is privacy enhancement in DOSN, through a user-friendly access control policies and a proper authentication. Our proposed solution is suitable for DOSN, specifically, when each user's profile is stored on his own computer. Each user's profile is considered as hierarchical blocks (depicted in figure 2.1) that are stored, together with related authorization policies in order to control access to these blocks (profile), on the a private machine. The policies are expressed in XACML, and every user uses a personal web server to enforce these access control policies. To authenticate the requesters, secret key based authentication method is selected to use in SAML. We assume that the secret key (used for authentication) is shared (preferably, out of band), between the requesters and owner, before sending access request. In this proposal, the interaction between the owner and requesters are depicted in figure 6.1.

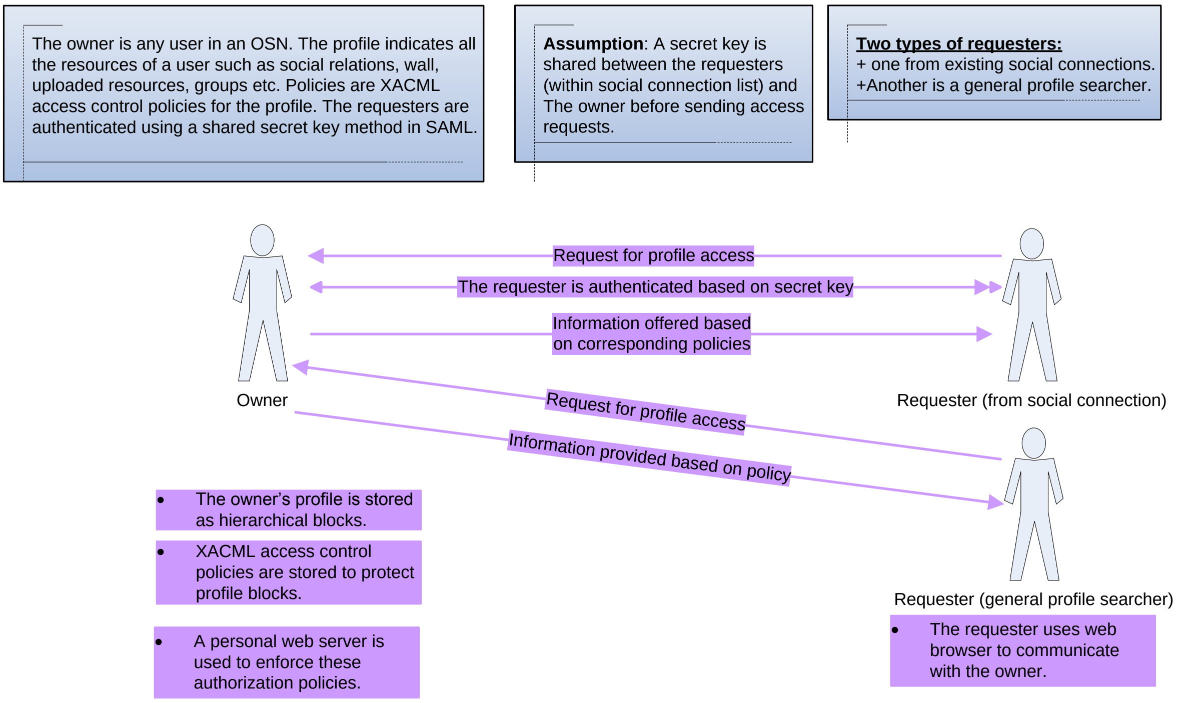

The owner is any user in an OSN. The profile indicates all the resources of a user such as social relations, wall, uploaded resources, groups etc. Policies are XACML access control policies for the profile. The requesters are authenticated using a shared secret key method in SAML.
Assumption: A secret key is shared between the requesters (within social connection list) and The owner before sending access requests.
Two types of requesters:
+ one from existing social connections.
+Another is a general profile searcher.
Request for profile access
The requester is authenticated based on secret key
Information offered based on corresponding policies
Owner
Requester (from social connection)
Request for profile access
Information provided based on policy
Requester (general profile searcher)
The owner's profile is stored as hierarchical blocks.
XACML access control policies are stored to protect profile blocks.
A personal web server is used to enforce these authorization policies.
The requester uses web browser to communicate with the owner.

Figure 6.1. Interaction between Owner and requesters in distributed online social network.

In our proposal, two types of requesters are considered. First, those that are from the owner's current social connections list, and other, those that are subscribers of the same OSN, but at that moment, not connected with the owner. The requesters, who are not from the current social list of the owner, are not required to be authenticated. However, it is mandatory for them to submit their email address or unique id (used in OSN) within access requests. Some general information from users' profiles (depending on user's settings) is available for all profile searchers.

Moreover, our solution is also applicable in a situation, where users store their profiles on other machines (replicas) but with the assumption that replicas are completely trusted. In other words, there is no extra security to maintain the confidentiality and integrity of these profiles on the replicas. In this case, the interaction among the owner, replicas and requesters is presented in the figure 6.2.

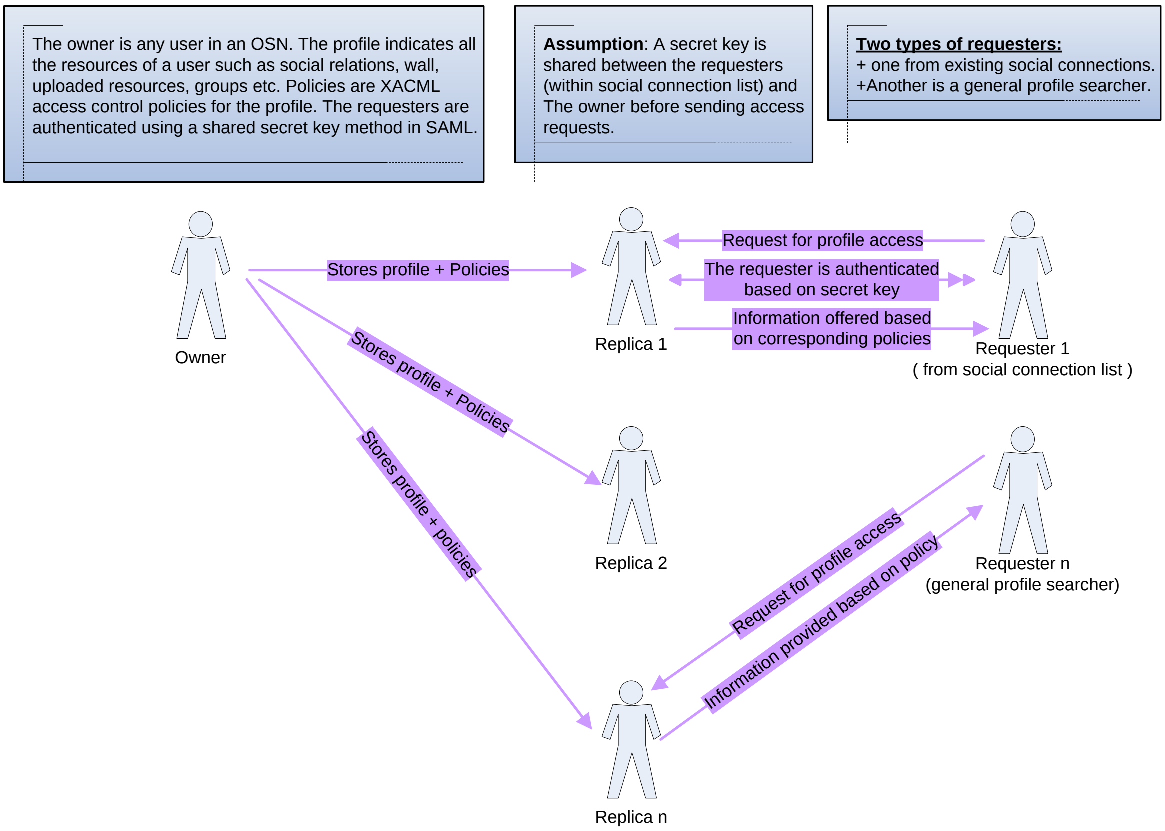

The owner is any user in an OSN. The profile indicates all the resources of a user such as social relations, wall, uploaded resources, groups etc. Policies are XACML access control policies for the profile. The requesters are authenticated using a shared secret key method in SAML.
Assumption: A secret key is shared between the requesters (within social connection list) and The owner before sending access requests.
Two types of requesters:
+ one from existing social connections.
+Another is a general profile searcher.
Owner
Stores profile + Policies
Stores profile + Policies
Stores profile + policies
Replica 1
Replica 2
Replica n
Request for profile access
The requester is authenticated based on secret key
Information offered based on corresponding policies
Requester 1
(from social connection list)
Requester n
(general profile searcher)
Request for profile access
Information provided based on policy

Figure 6.2. Interaction among Owner, replicas and requesters in distributed Social network.

The profile of a user, such as his/her private information, uploaded pictures, videos, or documents, created groups, or events, and social relationships list with separate attributes, is stored on the replicas. The questions related to this replication process, such as – how many replicas are selected to store a single profile, what are the criteria to select replicas, and what is the storage mechanism (whether its whole profile together or making small blocks from profile and then storing on replicas), are out of the scope of this thesis.

Therefore, in the whole interaction progress, when any requester (from current social list of the owner) sends his/her access request to any replica, at the first step, the requester is authenticated by using a shared secret key, which must be previously shared by the requester and owner (preferably shared in out-of-band). After the requester is authenticated successfully, the access decision (granted or denied) is evaluated and related information is returned to the requester based on the policies and the requester's attributes (from information within his/her request and also from the repository).

6.1.1 Reasons behind the Proposal

The main objective of our proposal is to protect users' information, both uploaded content (personal information, pictures, videos, events, messages, etc.) and relationship information (social connection graph, groups, browsing history etc.) from unauthorized access in DOSN. Our system is suitable for the OSNs, where all subscribers store their profiles in their own machines and use a personal web server to enforce access control policies on their profiles. Moreover, in the previous section, we present the idea, how to use our solution, where users' profiles are stored on replicas. These users' profiles contain not only sensitive information about the user only, but also hold some secret information about the people, who are socially connected with him. In this case, the replicas are responsible for authenticating requesters, and providing authorized resources. Therefore, we propose to store all the information about these social links in the replicas, and present the points elaborately, which support our idea.

Why should the whole social connections list and their attributes be stored on replicas?

Personal properties, such as how much a person trusts another person, or what is the exact relationship level with a particular user, etc. are very sensitive and have high impact on the social relationships, on disclosure of these attributes. Moreover, these attributes cannot considered as symmetric, therefore, one cannot say that A is highly trusted by B, only by knowing that B trusts A completely, or vice versa. Additionally, it is obvious that, if A treats B as close friend in OSN that does not ensure that B also has the same relationship with A. Therefore, these specific attributes vary from person to person in OSN, and completely, depends on the owner. And generally, on the basis of these attribute values an owner sets access privileges on his resources. Because of these reasons, we suggest storage of all these sensitive information regarding social connections on the replicas (the owner sets these properties of social relations), rather than on the requesters. Thus, when an access request comes to a replica, he can evaluate the corresponding policies, based on these stored attributed values without revealing it to the requester.

6.1.2 Our System and Different Possible Threats

Due to the increasing popularity of current OSNs, they have become prominent repository of users' private information in the form of profiles. Unfortunately, this factor has exploited several threats to the privacy of the users. Any user in OSN, mainly, suffers from five different kinds of attackers: OSN service provider; the users from his social link; users who are subscribers of the same OSN but not connected with that user; random person, who is behaviorally equivalent to an OSN subscriber but not socially connected; and application providers.

Among all, OSN service providers are the most powerful attacker because of underlying centralized client-server architecture. As OSN providers can access all uploaded information of their subscribers, hence, they can expose these data for different intensions. Providing data for targeted advertisements, or analyzing behavior pattern of their subscribers using data mining, are two widely known scenarios. However, p2p architecture for social networks removes the dependency from OSN providers by storing users' profiles on their private machines and therefore, users have complete control over their resources. Therefore, the absence of central authority removes the threats of these attackers.

Since different types of a third party applications are one of the main features in OSNs, application providers can use users' information (revealed during the time of using specific application) for malicious purposes without any knowledge of the information owner. Therefore, data the handing policies and preferences are two important parameters for designing complete privacy architecture for online social networks. But this type of attack is beyond the scope of this thesis.

Other attackers – users those that are either socially connected or not connected (but subscriber of same OSN) and also random attackers, are the main target of our proposed solution. In centralized architecture, we can define our access control conditions, and completely rely on OSN providers to enforce these conditions. But in a p2p architecture, it is a vital question - how do we prevent unauthorized access to the users' content? Our system considers an approximate list of access control requirements for p2p OSNs, and offers a mechanism, where users can define access control rules based on diverse types of constraints to ensure authorized access. Moreover, our solution considers the verification of the requesters' information within their access requests, and integrates secret key based authentication with authorization control. Therefore, it decreases the possibility of several threats, such as identity theft, or impersonation attack. Finally, our proposal to store each user's profile as several separate blocks, and authorization policies to protect these blocks, also enriches the privacy by:

- Hiding information about unauthorized resources from the users.
- Protecting social relationships information.

The users, who are socially connected with a user, have certain access rights based on their relationship with the owner. Therefore, at the beginning, it is required to prevent unauthorized accesses. These types of attackers are not powerful as OSN providers. But they may perform different types of attacks, especially when they are compromised or start behaves maliciously. Some different types of attacks and corresponding protections of our system are described below.

- **Unauthorized Access**: Any user starts behaving maliciously it's probably, because of relationship changes, which is very normal in real life. That's why, that malicious user may try to access some unapproved resources. But as we propose to use fine-grained access control policies for authorization control, and the malicious users are not able to change these authorization policies. Therefore, our system is successful in preventing unauthorized access over subscribers' resources.

- **Secondary Data collection**: Users sensitive information (which they upload as a part of profile) may be downloaded by some authorized users. The users from social connections list, who only have access to a certain part of the owner profile, but do not have download permission, can copy the authorized content without any knowledge of the owner. Our system does not offer any method to prevent users' profiles from unauthorized download.

The users, who are not socially connected with a user, have a general overview of users' profiles (depending on individual user's settings). Therefore, at the beginning, it is necessary to inform owners, who is searching for their profiles. For this reason, we propose to include email address or unique identifier within access requests, for this type of requesters. Again, we need to consider the random attackers, who are behaviorally equivalent to these types of users in OSNs. These types of attackers are not powerful as OSN providers, or socially connected to the users, but they can perform various types of attacks. Some possible attacks are described below:

- **Session Hijacking**: Our proposal is to authenticate a user, before evaluating his authorization permissions. The requester initiates a communication by sending his access request to the owner. Then the requester is authenticated by verifying his secret key, but we did not include any method to authenticate the owner. Therefore, this type of threat may violate our system completely.

- **Account Hacking**: Account hacking, is one of the major problems for current social network users. In our system, we do not propose how to handle these types of attacks. Nevertheless our proposal to save users' profiles on their own machines decreases the chance of hacking users' profiles. Moreover, the users will not need to depend on the central authority to delete their accounts (all uploaded contents, as a part of a profile).

- **Impersonation Attacks or Phishing Attacks**: To handle impersonation attacks or phishing attacks, we propose secret key based authentication of the requesters. Particularly, the secret key is shared between the requester and the owner in out of band. Moreover, as in our proposal the requesters initiate the interaction by sending access requests to the owner. Thereby, the possibility of most used phishing attack, using email is decreased. Again, the absence of central website for subscribers declines the possible phishing attacks, through the fake web sites.

- **Fake Profile**: Users' profiles are stored on their own machines and the access rights depend on their trust relationship with others in real life. The requesters are authenticated on previously shared secret key, and granting authorization is based on access control polices. Therefore, our system reduces the possibility of faking profiles.

- **Sybil Attacks:** This type of attack is the most common in p2p architectures. In this type of attack, a malicious user can create multiple fake identities. But in social networks, users can connect with each other based on social relationships and we propose to give access rights based on these relationships and trust level. Therefore, Sybil nodes cannot connect with the users without trust relationship, specially, when users do not accept any unknown person to their social connections list.

- **Private Information Leakage via friendship links**: In OSNs, generally, people are connected with each other through their real life relationships. Therefore, there is a possibility to predict individual's private information through their social relationships, which they don't want to disclose, such as location information, working place, etc. But in our system, we propose to treat the relationships list as a separate block and define access control policy to offer privacy to this block. Thus, only the authorized users can access the relationships information of a certain subscriber.

- **Private Information Leakage via group information:** Again, there is a possibility of leakage private information of the subscribers in an OSN, through their group information (open groups, no constraints to become members), which they don't want to share, such as political opinions, religious views, etc. In our system, we propose to control the access rights of different roles in a group by authorization policies, but do not offer any method, how to protect information of the members of open groups.

However, if we choose to store users' profiles on replicas, then our design choice, storing all information about social links in replicas, introduces different threats as well. Since change in relationship, or trust level between persons is quite common in social relationships, it is therefore, required to update the storage on replicas consistently. Moreover, because the access control policies are defined by setting constraints on these attributes (such as trust level, relationship type etc.), any inconsistences in these attribute values (in the replicas) may permit unauthorized access. Further, confirming synchronization of these dynamic attributes on all the replicas causes the problem of overhead traffic. Additionally, one more important question arises regarding the confidentiality and integrity of this sensitive information. In case of compromise of any replica or if any replica starts behaving maliciously, then, the communication between the requesters and any of the replica that have compromised, may reveal confidential information about the users. These kind of behavior effects not only the owner's social status, but also all the people, who are socially connected with him, and also their relationships.

6.2 Possible Extensions and Alternative Authentication Methods

In OSN, maximum availability of the users' profiles is one of the major requirements; therefore, to make our contribution more suitable in DOSN, we need to consider the situation, where users' profiles are stored on a number of replicas, rather than on their own machines. Selecting replicas based on real life trust relationships (assume, replicas are completely trusted) somehow, solves the problem of confidentiality and integrity of storage content. Using trust relationship for selecting replicas in OSNs, and storing users' profiles on these replicas in plain text or without any security enforcement is logical. Nevertheless, some other factors require careful consideration to confirm the secure environment, such as change in relationships, compromised replicas, etc. Although the elaborate study on these issues is left for the future research, we describe briefly some possible alternative approaches.

- **Symmetric key Cryptography**: In our proposal, we suggest to use secret keys to authenticate requesters, which are shared between the requesters and the owner before sending access requests. We can extend our proposal for replication (storage profiles), to support confidentiality of the storage content (in replicas), using encryption mechanism by encrypting profiles with the same secret (symmetric) key. Therefore, the replicas don't have any access to the storage profiles. For this purpose, XML encryption technique may be integrated with our current proposal. However, using encryption is extremely ineffective because of storage overhead problem. As the same content requires several times encryption, to ensure different access privileges for different users. Moreover, the traffic overhead increases since access rights in OSN for a particular user, changes with trust level between that user and a profile owner.

- **Credential/certificate-based authentication:** Another popular method, credential/certificate-based authentication, is a good choice to replace secret key based authentication in order to hide all the properties of social links from the replicas. We already mention some recent work about credential-based authentication and access control, and XACML extensions to support these features. This approach reduces the risk of disclosing information about the social relationships, particularly, when the replicas are compromised. Moreover, it reduces traffic overhead, as it does not require continuous update of dynamic properties of social links in replicas.

- **Anonymous communication:** Anonymous communication is required between requesters and replicas in order to make the social graphs of any user unpredictable to the replicas. We already identified the importance of protecting relationship information in OSNs, and our proposed structure for OSN profiles also supports this feature. Therefore, if the replicas can get the identity of the requesters (during the time of handling access requests), they can easily disclose some confidential information, not only about the users' but also about the people who are socially connected with them. To support anonymous communication, the concept of onion routing by Roger Dingledine et al. [50] can be used; its latest version is free to use, and almost independent from the underlying operating systems.

- **Selective attributes disclosure or proving conditions on attributes:** In addition, to enhance the privacy of the requesters' (social links of users), the current solution should be extended. Authorized requesters should be given access permit, only by disclosing required attributes, or satisfying conditions over their attributes, rather than include all the attributes within their access requests. We describe one recent work by Jan Camenisch et al. [34] in the language-based security chapter, on how to extend XACML to support this feature.

Chapter 7

Conclusion and Future Direction

This thesis is an attempt to integrate a privacy-enhanced access control mechanism with a p2p social network architecture, such as PeerSoN. We primarily concentrate on the expression of common access control policies for the users in online social networks. We offer an access control mechanism for p2p OSNs, based on two criteria, expressions of wide ranges of authorization conditions and full privacy support, such as authentication. We implemented these authorization policies in XACML [14] and combined with SAML [36] to confirm authenticity.

In the background section we have presented current research on different issues in p2p social networks. And in chapter two, we have shown why the current access control models and existing solutions are not suitable for use in distributed online social networks. We have proposed a profile structure for the users of DOSN where users do not have any knowledge about unauthorized resources. Offered structure enable users to control access to their resources based on social relationships or trust level. We have described diverse possible access control constraints in an OSN environment and implemented in XACML. To support the privacy features, we have selected a secret key based authentication method in SAML and also described elaborately the points why this method is more suitable in DOSN compared to others. Finally, we have stated the rationales behind our design choices and evaluated our proposal against different security threats.

In this thesis, we just take our first step to build a privacy-enhancing access control mechanism and incorporate with a p2p architecture of social network. Hence, the solution presented here is most suitable as a preliminary idea for detailed investigation towards security solutions in DOSN. Currently, our proposal is most suitable for the situation, where every user stores his profile on his own machine, or replicates in other machines, with the assumption that everyone is fully trusted. Therefore, as future work, we plan to focus on a solution where all the replicas are completely untrusted. This work introduces two main research questions, namely, the mechanism for storing access control policies securely and the maintenance of synchronization of access control policies in replicas. Furthermore, currently, we have suggested to use one of the open source web servers (supports XACML access control policies) to enforce authorization policies. But in the future, our plan is to extend the proposal to support access control enforcement in distributed environments.

Bibliography

[1]. K. Rzadca, A. Datta, S. Buchegger: *Replica Placement in P2P Storage: Complexity and Game Theoretic Analyses*, In proceedings of ICDCS 2010, Genoa, Italy, p. 599–609, June 2010.

[2]. S. Buchegger, D. Schiöberg, L. H. Vu, A. Datta: *PeerSoN: P2P Social Networking - Early Experiences and Insights*, In Proceedings of SocialNets 2009, The 2nd Workshop on Social Network Systems, Nuernberg, Germany, p. 46–52, March 2009.

[3]. I. Clarke, O. Sandberg, B. Wiley, T. W. Hong: *Freenet: A Distributed Anonymous Information Storage and Retrieval System*, Workshop on Design Issues in Anonymity and Unobservability, p. 46–67, 2000.

[4]. F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, I. Stoica: *Wide-Area Cooperative Storage with CFS*, In Proc. ACM SOSP'01, Banff, Canada, p. 38–52, October 2001.

[5]. A. Rowstron and P. Druschel: *Storage Management And Caching In PAST, A Large-Scale, Persistent Peer-To-Peer Storage Utility*, In Proc. ACM SOSP'01, Banff, Canada, p. 21-32,October 2001.

[6]. J. Kubiatowicz, D. Bindel, Y. Chen, S. Czerwinski, P. Eaton, D. Geels, R. Gummadi, S. Rhea, H. Weatherspoon, W. Weimer, C. Wells, B. Zhao: *OceanStore: An Architecture for Global-Scale Persistent Storage*, ACM ASPLOS, p. 190–201, 2000.

[7]. M. Landers, H. Zhang, K-L. Tan: *PeerStore : Better Performance by Relaxing in Peer-to-Peer Backup*, In proceedings of the Fourth International Conference on PeertoPeer Computing, p. 72–79, 2004.

[8]. C. Batten, K. Barr, A. Saraf, S. Treptin: *pStore: A secure peer-to-peer backup system*, Technical Memo MIT-LCS-TM-632, MIT Laboratory for Computer Science, p. 24–36, December 2001.

[9]. H. Jarrayaa, M. Laurent-Maknavicius: *A Secure Peer-to-Peer Backup Service Keeping Great Autonomy while under the Supervision of a Provider*, Computer and security, Volume 29, Number 2, p. 180-195, March 2010.

[10]. R. Narendula, T. G. Papaioannou, K. Aberer: *Privacy-aware and highly-available OSN profiles*, In proceedings of the 19th IEEE International Workshops on Enabling Technologies: Infrastructures for Collaborative Enterprises (WETICE 2010), p. 211–216, 2010.

[11]. L. H. Vu, K. Aberer, S. Buchegger, A. Datta: *Enabling Secure Secret Sharing in Distributed Online Social Networks*, In Proceedings of Annual Computer Security Applications Conference (ACSAC) 2009, Hawaii, December 7-11, p. 419–428, 2009.

[12]. C. A. Ardagna, J. Camenisch, M. Kohlweiss, R. Leenes, G. Neven, B. Priem, P. Samarati, D. Sommer, M. Verdicchio: *Exploiting Cryptography for Privacy-Enhanced Access Control A result of the PRIME Project*, Journal of Computer Security, IOS Press, p. 123–160, 2010.

[13]. Requirements and concepts for privacy enhancing access control in social networks and collaborative workspaces. Available at: *http://www.primelife.eu/results/ documents/H1.2.5 Requirements and concepts for privacy enhancing access control in social networks and collaborative workspaces.pdf.*

[14]. eXtensible Access Control Markup Language (XACML) Version 2.0, February 2005. Available at: *http://docs.oasis-open.org/xacml/2.0/access_control-xacml-2.0-core-spec-os.Pdf*.

[15]. C. A. Ardagna, S. D. C. di Vimercati, S. Paraboschi, E. Pedrini, P. Samarati: *An XACML-based privacy-centered access control system*, In proceedings of the first ACM workshop on Information security governance, Chicago, Illinois, USA, p. 49–58, 2009.

[16]. C. A. Ardagna, S. D. C. di Vimercati, G. Neven, S. Paraboschi, F-S. Preiss, P. Samarati, M. Verdicchio: *Enabling Privacy-preserving Credential-based Access Control with XACML and SAML*, 10th IEEE International Conference on Computer and Information Technology, p. 1090–1095, 2010.

[17]. W. Tolone, G-J. Ahn, T. Pai, S-P. Hong: *Access control in collaborative systems*, ACM Computing Surveys (CSUR), v.37 n.1, p. 29-41, March 2005.

[18]. B. Carminati, E. Ferrari, A. Perego: *Rule-based access control for social networks*, In on the Move to Meaningful Internet Systems 2006: OTM 2006 Workshops. LNCS, Springer-Verlag, p. 1734–1744, 2006.

[19]. J. Domingo-Ferrer: *A public-key protocol for social networks with private relationships*, Modeling Decisions for Artificial Intelligence, LNCS 4617, Springer-Verlag, p. 373–379, 2007.

[20]. F. Beato, M. Kohlweiss, K. Wouters: *Enforcing access control in social networks*, HotPets, 2009. Available at: *http://www.cosic.esat.kuleuven.be/publications/article-1240.pdf.*

[21]. M. S´anchez-Artigas, P. Garc´ıa-L´opez: *Pace: Privacy-protection for access control enforcement in p2p networks*, In Globe'09, p. 99–111, 2009.

[22]. C. Sturm, K. R. Dittrich, P. Ziegler: *An access control mechanism for P2P collaborations*, In proceedings of the 2008 international workshop on Data management in peer-to-peer systems, ACM: New York, NY, USA. p. 51—58, 2008.

[23]. E. Palomar, J. Estevez-Tapiador, J. Hernandez-Castro, A. Ribagorda: *Certificate-based access control in pure p2p networks*, In proceedings of the 6th International Conference on Peer-to-Peer Computing, IEEE, Cambridge, UK, p. 177–184, September 2006.

[24]. T. Wobber, T. L. Rodeheffer, D. B. Terry: *Policy-based Access Control for Peer-to-Peer Replication*, TechReport, MSR-TR-2009-15, Microsoft Research, 24 July 2009.

[25]. Lampson, W. Butler: *Protection*, In 5th Princeton Symposium on Information Science and Systems. P. 437–443. Reprinted in ACM Operate, Syst. Rev. 8, p. 18–24, 1974.

[26]. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, C. E. Youman: *Role-based access control models*, In IEEE Computer 29, p. 38–47, February 1996.

[27]. R. K. Thomas and R. S. Sandhu: *Conceptual foundations for a model of task-based authorizations*, In proceedings of 7th IEEE Computer Security Foundations Workshop, Franconia, P. 66–79, 1994.

[28]. R. K. Thomas and R. S. Sandhu: *Task-based authorization controls (TBAC): Models for active and enterprise-oriented authorization management*, In Database Security XI: Status and Prospects, T. Y. Lin and X. Qian, Eds. North-Holland, p. 166–181, 1997.

[29]. R. K. Thomas: *Team-based Access Control (TMAC): A Primitive for Applying Role-based Access Controls in Collaborative Environments*, In Proceedings of 2nd ACM Workshop on Role-Based Access Control, Fairfax, p. 13–19, 1997.

[30]. C. K. Georgiadis, I. Mavridis, G. Pangalos, R. K. Thomas: *Flexible Team-Based Access Control Using Contexts*, In proceedings of the sixth ACM symposium on Access control models and technologies, Chantilly, Virginia, United States, p. 21 – 27, 2001.

[31]. A. Bullock, S. Benford: *An access control framework for multi-user collaborative environments*, In proceedings of the international ACM SIGGROUP conference on Supporting group work, Phoenix, Arizona, United States, p. 140 – 149, 1999.

[32]. R. Baden, A. Bender, N. Spring, B. Bhattacharjee, D. Starin: *Persona: An Online Social Network with User-Defined Privacy*, In proceedings of ACM SIGCOMM, p. 135-146, August 2009.

[33]. J. Anderson, C. Diaz, J. Bonneau, F. Stajano: *Privacy-Enabling Social Networking Over Untrusted Networks*, In proceedings of the 2nd ACM workshop on Online social networks , Barcelona, Spain, p. 1-6 ,2009.

[34]. J. Camenisch, S. Modersheim, G. Neven, F-S. Preiss, D. Sommer: *Credential-Based Access Control Extensions to XACML*, W3C Workshop on Access Control Application Scenarios, Luxembourg, 17 and 18 November, 2009.

[35]. C. A. Ardagna, E. Pedrini, S. D. C. di Vimercati, P. Samarati, L. Bussard, G. Neven, F-S. Preiss, S. Paraboschi, M. Verdicchio, D. Raggett, S. Trabelsi: *PrimeLife Policy Language.*W3C Workshop on Access Control Application Scenarios, Luxembourg, 17 and 18 November, 2009.

[36]. Security Assertion Markup Language (SAML). Available at: http://xml.coverpages.org/ saml*html.*

[37]. SAML basics: A technical introduction to the Security Assertion Markup Language. Available at: *http://www2002.org/presentations/maler.pdf.*

[38]. How to Study and Learn SAML. Available at: *http://identitymeme.org/doc/draft-hodges-learning-saml-00.html.*

[39]. Bindings for the OASIS Security Assertion Markup Language (SAML) V2.0. Available at: *http://docs.oasis-open.org/security/saml/v2.0/saml-bindings-2.0-os.pdf.*

[40]. Security Assertion Markup Language (SAML) 2.0 Technical Overview, Working Draft 21, 21 February, 2007. Available at: *http://www.oasisopen.org/committees/download.php/22553/ sstc-saml-tech-overview-2%200-draft-13.pdf.*

[41]. SAML 2.0 profile of XACML v2.0. Available at: *http://docs.oasis-open.org/xacml/2.0/access _control-xacml-2.0-saml-profile-spec-os.pdf.*

[42]. Authentication. Available at: *http://www.cgisecurity.com/owasp/html/ch06.html.*

[43]. Authentication Context for the OASIS Security Assertion Markup Language (SAML) V2.0. Available at: *http://docs. oasis-open.org/security/saml/v2.0/saml-authn-context-2.0-os.pdf*

[44]. T.Wu: *The SRP Authentication and Key Exchange System*. Available at: *http://tools.ietf.org /html /rfc2945*.

[45].The Factors of Authentication. Available at: *http://www.e.govt.nz/standards/ authentication/guidance-on-multi-factor-authentication/factors-of-authentication.*

[46]. M. C. Mont, S. Pearson, P. Bramhall: *Towards accountable management of identity and privacy: Sticky policies and enforceable tracing services*, In proceedings of the 14th International Workshop on Database and Expert Systems Applications, p. 377-382, 2003.

[47]. Hierarchical resource profile of XACML v2.0. Available at: *http://docs.oasis-open.org /xacml/2.0/access_control-xacml-2.0-hier-profile-spec-os.pdf.*

[48]. XACML Profile for Role Based Access Control (RBAC). Available at: *http://docs.oasis-open .org/xacml/cd-xacml-rbac-profile-01.pdf.*

[49]. Social Networks Sites: Definition and History. Available at: *http://www.wealth-center.com/main/component/content/article/1-advertising/4-socialnetworking.html*.

[50]. R. Dingledine, N. Mathewson, P. Syverson: *Tor: The Second-Generation Onion Router*, In proceedings of the 13th USENIX Security Symposium, p. 303-320, August 2004.

[51]. Youssef Afify: *Access Control in a Peer-to-peer Social Network*. Master's Thesis, EPFL, Lausanne, Switzerland, August 15, 2008. Available at: http://www.peerson.net/papers/ Rapport.pdf

Appendix A

Access Control Policies

A.1 XACML Policy for “Scenario1”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                        access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
        Policy for scenario 1 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:rule"
     Effect="Permit">
    <Description>
         Karim can view the pictures of the album "After-Exam-Party" from Rahim's profile.
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal">
            <AttributeValue
               DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
                           karim@yahoo.com</AttributeValue>
            <SubjectAttributeDesignator
               SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
               AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               MustBePresent="true"
               DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                         file://profile/Rahim/Pictures/Album/After-Exam-Party </AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```

A.2 XACML Policy for “Scenario1-Extension1”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 1 extension 1 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:rule"
     Effect="Permit">
    <Description>
        Karim can view the pictures of the album "After-Exam-Party" from Rahim's profile.
    </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Pictures/Album/After-Exam-Party</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal">
        <AttributeValue
           DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
                       karim@yahoo.com</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only">
       <SubjectAttributeDesignator
         SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
         DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
        </Apply></Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal">
        <AttributeValue
           DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
                      karim_bd@gmail.com</AttributeValue>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only">
          <SubjectAttributeDesignator
          SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
        </Apply>
        </Apply>
    </Apply> </Condition>
  </Rule>
</Policy>
```

A.3 XACML Policy for “Scenario1-Extension2”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1ex2:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 1 extension 2 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1ex2:rule"
     Effect="Permit">
    <Description>
        Karim can view the pictures of the album "After-Exam-Party" from Rahim's profile.
    </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#anyURI">
              file://profile/Rahim/Pictures/Album/After-Exam-Party</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">tag</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal">
        <AttributeValue
           DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
                       karim@yahoo.com</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only">
       <SubjectAttributeDesignator
         SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
         DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
         </Apply></Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-equal">
        <AttributeValue
           DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">
                       karim_bd@gmail.com</AttributeValue>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:rfc822Name-one-and-only">
          <SubjectAttributeDesignator
          SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
```

```
          AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
          DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
        </Apply>
        </Apply>
    </Apply> </Condition>
  </Rule>
</Policy>
```

A.4 XACML Policy for “Scenario2”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario2:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 2 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario2:rule"
     Effect="Permit">
    <Description>
        Karim can view the video tiled "Birthday_Celebration" from Rahim's profile.
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">High</AttributeValue>
            <SubjectAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:scenario2:attribute:trust_level"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Videos/Birthday_Celebration</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```

A.5 XACML Policy for “Scenario3”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 3 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario1:rule"
     Effect="Permit">
    <Description>
        Karim can view the document titled "Weekend_parties" from Rahim's profile.
    </Description>
    <Target>
      <Subjects>
        <Subject>
          <SubjectMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">friends</AttributeValue>
            <SubjectAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:scenario2:attribute:relations"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
      </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Documents/Weekend_parties</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
  </Rule>
</Policy>
```

A.6 XACML Policy for “Scenario4”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario4:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
     Policy for scenario 4 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario4:rule"
     Effect="Permit">
    <Description>
        Karim can view and comment on the document titled "bidding_quote" from Rahim's profile.
    </Description>
    <Target>
      <Subjects>
       <Subject>
        </Subject>
       </Subjects>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Documents/bidding_quote</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">High</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
       <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:scenario4:attribute:trust_level"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply></Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">colleagues</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
       <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:scenario4:attribute:relations"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply></Apply>
</Apply> </Condition>
  </Rule>
</Policy>
```

A.7 XACML Policy for “Scenario5”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario5:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 5 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario5:rule"
     Effect="Permit">
```

```
<Description>
     Aybody form family or friends list of Rahim can view and comment in the pictures of the Album
    titled "study_abroad" from Rahim's profile.
</Description>
<Target>
  <Resources>
    <Resource>
      <ResourceMatch
         MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                      file://profile/Rahim/Pictures/Album/study_abroad</AttributeValue>
        <ResourceAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
           DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
      </ResourceMatch>
    </Resource>
  </Resources>
  <Actions>
    <Action>
      <ActionMatch
         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
        <ActionAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
           DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </ActionMatch>
    </Action>
    <Action>
      <ActionMatch
         MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
        <ActionAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
           DataType="http://www.w3.org/2001/XMLSchema#string"/>
      </ActionMatch>
    </Action>
  </Actions>
</Target>
<Condition>
 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">friends</AttributeValue>
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
   <SubjectAttributeDesignator
     AttributeId="urn:oasis:names:tc:xacml:1.0:scenario4:attribute:relations"
     DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Apply></Apply>
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
    <AttributeValue
       DataType="http://www.w3.org/2001/XMLSchema#string">family</AttributeValue>
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
   <SubjectAttributeDesignator
     AttributeId="urn:oasis:names:tc:xacml:1.0:scenario4:attribute:relations"
     DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </Apply></Apply>
```

```
</Apply> </Condition>
  </Rule>
</Policy>
```

A.8 XACML Policy for "Scenario6"

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                        access_control-xacml-2.0-policy-schema-os.xsd"
    PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario6:policy"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 6 presented in the document.
  </Description>
  <Target/>
  <Rule
      RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario6:rule"
      Effect="Permit">
    <Description>
        Aybody form social connection list of Rahim can view and comment in the document titled
        "National_day_Event" before 24th December 2010.
    </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                        file://profile/Rahim/Documents/National_day_Event</AttributeValue>
            <ResourceAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
            <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
        <EnvironmentAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"
           DataType="http://www.w3.org/2001/XMLSchema#date"/>
      </Apply>
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#date">2010-12-24</AttributeValue>
</Apply></Condition>
  </Rule>
</Policy>
```

A.9 XACML Policy for “Scenario7”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario7:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
    Policy for scenario 7 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario7:rule"
     Effect="Permit">
    <Description>
         Anybody invited by Rahim can view and comment in the group page named "Security requirements
        in social network" between 6.01 am and 5.59 pm (GMT time).
    </Description>
    <Target>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                  file://profile/Rahim/Group_pages/Security_requirements_in_social_network
               </AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
```

```
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
      <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
        </Apply>
     <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#time">06:01:00</AttributeValue>
    </Apply>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal">
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
      <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
          AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
     </Apply>
     <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#time">17:59:00</AttributeValue>
    </Apply>
   </Apply></Condition>
  </Rule>
</Policy>
```

A.10 XACML Policy for "Scenario8"

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario8:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-unless-deny">
  <Description>
      Policy for scenario 8 presented in the document.
  </Description>
  <Target>
      <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
```

```
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Group_pages/Information_Zone</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
   <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario8:rule1"
     Effect="Permit">
    <Description>
        All the colleagues of Rahim can access the group page titled "Information Zone" and share
        their ideas as comment.
    </Description>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">colleagues</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
       <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:scenario8:attribute:relations"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
     </Apply></Condition>
</Rule>
<Rule RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario8:rule2"
   Effect="Permit">
    <Description>
       Access the group page titled "Information Zone" is only possible during 8am to 5pm.
    </Description>
    <Target/>
    <Condition>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-greater-than-or-equal">
```

```
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
      <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
             AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
        </Apply>
           <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">
                                        08:00:00</AttributeValue>
     </Apply>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-less-than-or-equal">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:time-one-and-only">
       <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#time"
           AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-time"/>
      </Apply>
      <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#time">
                                  17:00:00</AttributeValue>
     </Apply>
    </Apply></Condition>
</Rule>
<Rule RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario8:rule3" Effect="Deny">
    <Description>
        Access the group page titled "Information Zone" is only possible during working days
        that is from Monday to Friday.
     </Description>
     <Target/>
  <Condition>
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Saturday</AttributeValue>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
        <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:2.0:scenario8:attribute:dayOfWeek"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </Apply></Apply>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
         <AttributeValue
            DataType="http://www.w3.org/2001/XMLSchema#string">Sunday</AttributeValue>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
        <SubjectAttributeDesignator
          AttributeId="urn:oasis:names:tc:xacml:2.0:scenario8:attribute:dayOfWeek"
          DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </Apply></Apply>
  </Apply>
 </Condition>
 </Rule>
</Policy>
```

A.11 XACML Policy for “Scenario9”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                             access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario9:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 9 presented in the document.
```

```
</Description>
<Target/>
<Rule
   RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario9:rule"
   Effect="Permit">
  <Description>
       Anybody from social connection of Rahim has age more than 22 during the time of his request,
      can view and comment to the video titled "Supernatural_Activity" from Rahim's profile
  </Description>
  <Target>
    <Resources>
      <Resource>
        <ResourceMatch
           MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
          <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                         file://profile/Rahim/Videos/Supernatural_Activity</AttributeValue>
          <ResourceAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
             DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
        </ResourceMatch>
      </Resource>
    </Resources>
    <Actions>
      <Action>
        <ActionMatch
           MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
          <ActionAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
             DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
      </Action>
      <Action>
        <ActionMatch
           MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
          <ActionAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
             DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </ActionMatch>
      </Action>
    </Actions>
  </Target>
  <Condition>
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
    <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
        <SubjectAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:2.0:scenario9:attribute:date_of_birth"
           DataType="http://www.w3.org/2001/XMLSchema#date"/>
      </Apply>
      <AttributeValue
         DataType="http://www.w3.org/TR/2002/WD-xquery-operators-
                    20020816#yearMonthDuration">P21Y12M</AttributeValue>
    </Apply>
```

```
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
      <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
          AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"/>
      </Apply>
     </Apply>
   </Condition>
  </Rule>
</Policy>
```

A.12 XACML Policy for “Scenario10”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario10:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 10 presented in the document.
  </Description>
  <Target/>
  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario10:rule"
     Effect="Permit">
    <Description>
        Karim or Jishan can view the pictures of the album "Life in a Forest" and/or the video "Nature of
        Wild Animal" from Rahim's profile.
    </Description>
    <Target>
     <Subjects>
        <Subject>
          <SubjectMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">
                           KarimHasnat</AttributeValue>
            <SubjectAttributeDesignator
               SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
               AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               Issuer="http://www.Hunting-tecniques.com/club-authority"
               MustBePresent="true"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
        <Subject>
          <SubjectMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">
                           Jishan Hasib</AttributeValue>
            <SubjectAttributeDesignator
               SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
               AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
               Issuer="http://www.Hunting-tecniques.com/club-authority"
               MustBePresent="true"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </SubjectMatch>
        </Subject>
    </Subjects>
    <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
              file://profile/Rahim/Pictures/Album/Life-In-a-Forest</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           file://profile/Rahim/Videos/Nature-of-Wild-Animal</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
     </Target>
 </Rule>
</Policy>
```

A.13 XACML Policy for “Scenario11”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario11:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 11 presented in the document.
  </Description>
  <Target/>
<Rule
```

```
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario11:rule"
     Effect="Permit">
    <Description>
        Any person request for a document by providing the keyword "sweden" as a resource id is able
        to access the document "Before coming to Sweden" from Rahim's profile.
    </Description>
    <Target>
      <Actions>
        <Action>
          <ActionMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
            <ActionAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
          </ActionMatch>
        </Action>
      </Actions>
    </Target>
 <Condition>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">.* Sweden</AttributeValue>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <ResourceAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
             DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
        <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">Sweden .*</AttributeValue>
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
          <ResourceAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
             DataType="http://www.w3.org/2001/XMLSchema#string"/>
                  </Apply>
      </Apply>
  </Apply></Condition>
  </Rule>
</Policy>
```

A.14 XACML Policy for "Scenario11-Extension1"

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario11ex1:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
     Policy for scenario.11 extension.1 presented in the document.
  </Description>
  <Target/>
```

```
 <Rule
    RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario11ex1:rule"
    Effect="Permit">
   <Description>
       Any person request for a document by providing the keyword "Sweden" as a resource id and
       "current" and "old" as version infromation is able to access the document "Before coming to
        Sweden" from Rahim's profile.
   </Description>
  <Target>
   <Resources>
      <Resource>
         <ResourceMatch
            MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">current</AttributeValue>
           <ResourceAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:2.0:scenario11ex1:attribute:version_information"
              MustBePresent="true"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ResourceMatch>
      </Resource>
      <Resource>
         <ResourceMatch
            MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">old</AttributeValue>
           <ResourceAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:2.0:scenario11ex1:attribute:version_information"
              MustBePresent="true"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ResourceMatch>
      </Resource>
    </Resources>
     <Actions>
       <Action>
         <ActionMatch
            MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
           <ActionAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ActionMatch>
       </Action>
     </Actions>
  </Target>
<Condition>
 <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
       <AttributeValue
          DataType="http://www.w3.org/2001/XMLSchema#string">.* Sweden</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
         <ResourceAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
            DataType="http://www.w3.org/2001/XMLSchema#string"/>
       </Apply>
     </Apply>
     <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-regexp-match">
       <AttributeValue
```

```
            DataType="http://www.w3.org/2001/XMLSchema#string">Sweden .*</AttributeValue>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
           <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#string"/>
                 </Apply>
       </Apply>
      </Apply>
    </Condition>
  </Rule>
 </Policy>
```

A.15 XACML Policy for “Scenario12”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
    xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                              access_control-xacml-2.0-policy-schema-os.xsd"
    PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario12:policy"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
       Policy for scenario 12 presented in the document.
  </Description>
  <Target/>
   <Rule
      RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario12:rule"
      Effect="Permit">
    <Description>
         Anybody from Rahim's friends list and have a high trust relationship with Rahim is able to
         view and comment in all the uploaded albums from Rahim's profile.
    </Description>
   <Target>
    <Resources>
         <Resource>
           <ResourceMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
             <AttributeValue
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                             urn:profile:Rahim:pictures</AttributeValue>
             <ResourceAttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
           </ResourceMatch>
         </Resource>
         <Resource>
           <ResourceMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
             <AttributeValue
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                             urn:profile:Rahim:pictures:After-Exam-Party</AttributeValue>
             <ResourceAttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
           </ResourceMatch>
         </Resource>
         <Resource>
    <ResourceMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                     urn:profile:Rahim:pictures:Life-In-a-Forest</AttributeValue>
      <ResourceAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
         DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                     urn:profile:Rahim:pictures:Study-Abroad</AttributeValue>
      <ResourceAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
         DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                  urn:profile:Rahim:pictures:After-Exam-Party:After-first-sem</AttributeValue>
      <ResourceAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
         DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ResourceMatch>
  </Resource>
  <Resource>
    <ResourceMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                 urn:profile:Rahim:pictures:After-Exam-Party:After-second-sem</AttributeValue>
      <ResourceAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
         DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
    </ResourceMatch>
  </Resource>
 </Resources>
<Actions>
  <Action>
    <ActionMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
         DataType="http://www.w3.org/2001/XMLSchema#string">view</AttributeValue>
      <ActionAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
    </ActionMatch>
  </Action>
  <Action>
    <ActionMatch
       MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
      <AttributeValue
```

```
                DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
             <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
           </ActionMatch>
         </Action>
       </Actions>
     </Target>
    <Condition>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">High</AttributeValue>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
         <SubjectAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:scenario12:attribute:trust_level"
           DataType="http://www.w3.org/2001/XMLSchema#string"/>
           </Apply></Apply>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
          <AttributeValue
             DataType="http://www.w3.org/2001/XMLSchema#string">friends</AttributeValue>
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
         <SubjectAttributeDesignator
           AttributeId="urn:oasis:names:tc:xacml:1.0:scenario12:attribute:relations"
           DataType="http://www.w3.org/2001/XMLSchema#string"/>
           </Apply></Apply>
    </Apply> </Condition>
  </Rule>
</Policy>
```

A.16 XACML Policy for “Scenario13”

Access privileges for the role “Administrators”:

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                                    access_control-xacml-2.0-policySet-schema-os.xsd"
        PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:Administrators:role"
        PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-
overrides">
   <Description>
         Permission policy set for the role "Administrators" from scenario 13 presented in the document.
   </Description>
   <Target/>

   <Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                     access_control-xacml-2.0-policy-schema-os.xsd"
        PolicyId="urn:oasis:names:tc:xacml:2.0:scenario13:Permissions:Administrators:role"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

<Rule RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario13:rule1"
        Effect="Permit">
    <Target>
      <Resources>
```

```
          <Resource>
               <ResourceMatch
                  MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
                 <AttributeValue
                    DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                                  urn:profile:Jishan:group:P2P-Security</AttributeValue>
                 <ResourceAttributeDesignator
                    AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                    DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
               </ResourceMatch>
            </Resource>
        </Resources>
        <Actions>
         <Action>
            <ActionMatch
                  MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                 <AttributeValue
                    DataType="http://www.w3.org/2001/XMLSchema#string">join</AttributeValue>
                 <ActionAttributeDesignator
                    AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                    DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ActionMatch>
         </Action>
         <Action>
            <ActionMatch
                  MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
                 <AttributeValue
                    DataType="http://www.w3.org/2001/XMLSchema#string">discard</AttributeValue>
                 <ActionAttributeDesignator
                    AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                    DataType="http://www.w3.org/2001/XMLSchema#string"/>
            </ActionMatch>
         </Action>
        </Actions>
       </Target>
     </Rule>
    </Policy>
  <PolicySetIdReference>
     urn:oasis:names:tc:xacml:2.0:scenario13:ProfessionalMembers:role</PolicySetIdReference>
  <PolicySetIdReference>
     urn:oasis:names:tc:xacml:2.0:scenario13:OrdinaryMembers:role</PolicySetIdReference>
</PolicySet>
```

Access privileges for the role “Professional Members”:

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                    access_control-xacml-2.0-policySet-schema-os.xsd"
        PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:ProfessionalMembers:role"
        PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-
overrides">
   <Description>
         Permission policy set for the role "Professional Members" from scenario 13 presented in the docu-
         ment.
   </Description>
   <Target/>
```

```
<Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                                  access_control-xacml-2.0-policy-schema-os.xsd"
        PolicyId="urn:oasis:names:tc:xacml:2.0:scenario13:Permissions:ProfessionalMembers:role"
        RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

<Rule RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario13:rule2"
       Effect="Permit">
    <Target>
      <Resources>
        <Resource>
            <ResourceMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
              <AttributeValue
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                            urn:profile:Jishan:group:P2P-Security</AttributeValue>
              <ResourceAttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                 DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
            </ResourceMatch>
        </Resource>
      </Resources>
      <Actions>
       <Action>
          <ActionMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue
                 DataType="http://www.w3.org/2001/XMLSchema#string">modify</AttributeValue>
              <ActionAttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                 DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ActionMatch>
       </Action>
       <Action>
          <ActionMatch
               MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
              <AttributeValue
                 DataType="http://www.w3.org/2001/XMLSchema#string">comment</AttributeValue>
              <ActionAttributeDesignator
                 AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                 DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </ActionMatch>
       </Action>
      </Actions>
    </Target>
   </Rule>
  </Policy>
 <PolicySetIdReference>urn:oasis:names:tc:xacml:2.0:scenario13:OrdinaryMembers:role
  </PolicySetIdReference>
</PolicySet>
```

Access privileges for the role “Ordinary Members”:

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                                  access_control-xacml-2.0-policySet-schema-os.xsd"
```

```
      PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:OrdinaryMembers:role"
    PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
  <Description>
       Permission policy set for the role "Ordinary Members" from scenario 13 presented in the document.
  </Description>
  <Target/>

 <Policy xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                                access_control-xacml-2.0-policy-schema-os.xsd"
       PolicyId="urn:oasis:names:tc:xacml:2.0:scenario13:Permissions:OrdinaryMembers:role"
       RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

  <Rule RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario13:rule3"
      Effect="Permit">
    <Target>
     <Resources>
       <Resource>
           <ResourceMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
             <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                           urn:profile:Jishan:group:P2P-Security</AttributeValue>
             <ResourceAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
                DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
           </ResourceMatch>
         </Resource>
     </Resources>
     <Actions>
      <Action>
        <ActionMatch
              MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
             <AttributeValue
                DataType="http://www.w3.org/2001/XMLSchema#string">read</AttributeValue>
             <ActionAttributeDesignator
                AttributeId="urn:oasis:names:tc:xacml:1.0:action:action-id"
                DataType="http://www.w3.org/2001/XMLSchema#string"/>
       </ActionMatch>
      </Action>
     </Actions>
    </Target>
  </Rule>
 </Policy>
</PolicySet>
```

RolePolicy for the role “Administrators”:

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                                access_control-xacml-2.0-policySet-schema-os.xsd"
       PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:Role:Administrators:role"
    PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
 <Description>
      Role policy set for the role "Administrators" from scenario-13 presented in the document.
 </Description>
```

```
  <Target>
    <Subjects>
       <Subject>
            <SubjectMatch
                MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
                   DataType="http://www.w3.org/2001/XMLSchema#string">
                                 Administrators</AttributeValue>
               <SubjectAttributeDesignator
                   AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
                   DataType="http://www.w3.org/2001/XMLSchema#string"/>
             </SubjectMatch>
       </Subject>
     </Subjects>
  </Target>

<PolicySetIdReference>urn:oasis:names:tc:xacml:2.0:scenario13:Administrators:role
   </PolicySetIdReference>
</PolicySet>
```

RolePolicy for the role "Professional Members":

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                                  access_control-xacml-2.0-policySet-schema-os.xsd"
       PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:Role:ProfessionalMembers:role"
    PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
   <Description>
        Role policy set for the role "Professional Members" from scenario-13 presented in the document.
   </Description>
  <Target>
    <Subjects>
       <Subject>
            <SubjectMatch
                MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
               <AttributeValue
                   DataType="http://www.w3.org/2001/XMLSchema#string">
                                  Professionals</AttributeValue>
               <SubjectAttributeDesignator
                   AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
                   DataType="http://www.w3.org/2001/XMLSchema#string"/>
             </SubjectMatch>
       </Subject>
     </Subjects>
  </Target>
<PolicySetIdReference>urn:oasis:names:tc:xacml:2.0:scenario13:ProfessionalMembers:role
   </PolicySetIdReference>
</PolicySet>
```

RolePolicy for the role "Ordinary Members":

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
```

```
                                   access_control-xacml-2.0-policySet-schema-os.xsd"
    PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:Role:OrdinaryMembers:role"
  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
  <Description>
      Role policy set for the role "Ordinary Members" from scenario-13 presented in the document.
  </Description>
 <Target>
   <Subjects>
     <Subject>
         <SubjectMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
           <AttributeValue
              DataType="http://www.w3.org/2001/XMLSchema#string">Members</AttributeValue>
           <SubjectAttributeDesignator
              AttributeId="urn:oasis:names:tc:xacml:2.0:subject:role"
              DataType="http://www.w3.org/2001/XMLSchema#string"/>
         </SubjectMatch>
     </Subject>
   </Subjects>
  </Target>
 <PolicySetIdReference>urn:oasis:names:tc:xacml:2.0:scenario13:OrdinaryMembers:role
     </PolicySetIdReference>
</PolicySet>
```

A.17 XACML Policy for "Scenario13-Extension1"

RolePolicy for the role "Professional Members":

```
<?xml version="1.0" encoding="UTF-8"?>
<PolicySet xmlns="urn:oasis:names:tc:xacml:2.0:policySet:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policySet:schema:os
                                 access_control-xacml-2.0-policySet-schema-os.xsd"
    PolicySetId="urn:oasis:names:tc:xacml:2.0:scenario13:Role:ProfessionalMembers:role"
  PolicyCombiningAlgId="urn:oasis:names:tc:xacml:1.0:policy-combining-algorithm:permit-overrides">
  <Description>
   Role policy set for the role "Professional Members" from scenario-13-extension 1 presented.
   in the document.
  </Description>
  <Target/>

<Policy
  xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
              access_control-xacml-2.0-policy-schema-os.xsd"
    PolicyId="urn:oasis:names:tc:xacml:2.0:scenario13:Permissions:Administrators:role"
    RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:permit-overrides">

<Rule
    RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario13:rule4"
    Effect="Permit">
   <Target/>

 <Condition>
  <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:or">
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
```

```
           DataType="http://www.w3.org/2001/XMLSchema#string">Professionals</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
       <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:scenario13:attribute:role"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply></Apply>
   <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:and">
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-equal">
       <AttributeValue
           DataType="http://www.w3.org/2001/XMLSchema#string">Members</AttributeValue>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:string-one-and-only">
       <SubjectAttributeDesignator
         AttributeId="urn:oasis:names:tc:xacml:1.0:scenario13:attribute:role"
         DataType="http://www.w3.org/2001/XMLSchema#string"/>
        </Apply></Apply>
       <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-less-than-or-equal">
         <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-add-yearMonthDuration">
        <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
          <SubjectAttributeDesignator
             AttributeId="urn:oasis:names:tc:xacml:2.0:scenario13:attribute:date_of_join"
             DataType="http://www.w3.org/2001/XMLSchema#date"/>
        </Apply>
        <AttributeValue
        DataType="http://www.w3.org/TR/2002/WD-xquery-operators-
                   20020816#yearMonthDuration">P1Y12M</AttributeValue>
      </Apply>
      <Apply FunctionId="urn:oasis:names:tc:xacml:1.0:function:date-one-and-only">
      <EnvironmentAttributeDesignator DataType="http://www.w3.org/2001/XMLSchema#date"
          AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-date"/>
      </Apply>
    </Apply></Apply>
   </Apply></Condition>
  </Rule>
 </Policy>
<PolicySetIdReference>urn:oasis:names:tc:xacml:2.0:scenario13:ProfessionalMembers:role
   </PolicySetIdReference>
</PolicySet>
```

A.18 XACML Policy for “Scenario14”

```
<?xml version="1.0" encoding="UTF-8"?>
<Policy
   xmlns="urn:oasis:names:tc:xacml:2.0:policy:schema:os"
   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
   xsi:schemaLocation="urn:oasis:names:tc:xacml:2.0:policy:schema:os
                         access_control-xacml-2.0-policy-schema-os.xsd"
   PolicyId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario14:policy"
   RuleCombiningAlgId="urn:oasis:names:tc:xacml:1.0:rule-combining-algorithm:deny-overrides">
  <Description>
      Policy for scenario 14 presented in the document.
  </Description>
  <Target/>

  <Rule
     RuleId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario14:rule"
     Effect="Permit">
    <Description>
```

```
        Anybody search for Rahim's profile is able to view the basic information from Rahim's profile
       page.
    </Description>
  <Target>
    <Resources>
        <Resource>
          <ResourceMatch
             MatchId="urn:oasis:names:tc:xacml:1.0:function:anyURI-equal">
            <AttributeValue
               DataType="http://www.w3.org/2001/XMLSchema#anyURI">
                          file://profile/Rahim/Basic_Information</AttributeValue>
            <ResourceAttributeDesignator
               AttributeId="urn:oasis:names:tc:xacml:1.0:resource:resource-id"
               DataType="http://www.w3.org/2001/XMLSchema#anyURI"/>
          </ResourceMatch>
        </Resource>
    </Resources>
  </Target>
</Rule>
<Obligations>
    <Obligation
      ObligationId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:Scenario14:emailNotification"
      FulfillOn="Permit">
       <AttributeAssignment
        AttributeId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:emailrecipient"
        DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name">rahim@yahoo.com
       </AttributeAssignment>
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:emailbody"
          DataType="http://www.w3.org/2001/XMLSchema#string">
        <AttributeValue DataType="http://www.w3.org/2001/XMLSchema#string">
           Rahim, your profile is searched and accessed by:
        </AttributeValue>
      </AttributeAssignment>
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:requesterId"
          DataType="http://www.w3.org/2001/XMLSchema#string">
       <SubjectAttributeDesignator
         SubjectCategory="urn:oasis:names:tc:xacml:1.0:subject-category:access-subject"
         AttributeId="urn:oasis:names:tc:xacml:1.0:subject:subject-id"
         MustBePresent="true"
         DataType="urn:oasis:names:tc:xacml:1.0:data-type:rfc822Name"/>
            </AttributeAssignment>
      <AttributeAssignment
          AttributeId="urn:oasis:names:tc:xacml:2.0:XACML-IN-OSN:accessedtime"
          DataType="http://www.w3.org/2001/XMLSchema#string">
      <EnvironmentAttributeDesignator
            AttributeId="urn:oasis:names:tc:xacml:1.0:environment:current-dateTime"
            MustBePresent="true"
            DataType="http://www.w3.org/2001/XMLSchema#dateTime"/>
      </AttributeAssignment>
    </Obligation>
</Obligations>
</Policy>
```