IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 3, Ver. I (May. - June. 2018), PP 27-33
www.iosrjournals.org

Privacy in Distributed Access Control Environment

U. M. Mbanaso,
Ph.D. Centre for Cyberspace Studies, Nasarawa State University, Keffi, Nigeria.
Corresponding Author: U. M. Mbanaso

Abstract: *Privacy in distributed environments introduces new security challenges since two parties in a typical access control may not always belong to the same security domain. Therefore, protection of personal identifiable information (PII) throws a fresh challenge in privacy equation in this context. Customarily, the challenge is how to preserve privacy when two parties in different security domains are involved in access control, particularly in distributed environments. This paper presents the conceptualization of privacy in typical distributed context, to raise issues when two parties in access control requires to protect privacy sensitive information.*

Date of Submission: 05-05-2018 Date of acceptance: 26-05-2018

I. Introduction

The web technology has somewhat changed the way Information Systems (IS) interact today; as a consequence, traditional approaches to system development[1] have altered as well. Driven by these new business environments, organizations have a need to expose some part of their enterprise services to the outside world. This implies greater demand to allow access to enterprise computing operations whilst ensuring security and privacy. Addressing privacy and security issues in this context necessitates a requirements gathering for the application development phase. Given the multiple level of interactions between participating parties in distributed electronic transactions, requirements analysis is a critical phase, which covers the complex tasks of deconstructing how a system should behave, as well as helping to expose the system properties, or attributes [2]. Thus, the need for a thorough system requirements gathering to guide the software development life cycle phases is imperative.

This paper examines the generalised view of distributed access control, and briefly reviews authentication, authorization and trust in the context of the distributed XACML model. It describes a security threat model based on a typical e-procurement use-case in an attempt to define the scope of the applicable privacy and security requirements. This systematic approach is expected to aid in identifying the principal actors, and their relationships in distributed access control interactions. Furthermore, the paper will also appraise trust models and various options for establishing trust relationships in distributed environments. Additionally, the paperdeals with the policy framework, detailed description of the WS-XACML profile, architecture and usage. Lastly, a conceptual design supporting this work resulting from the inputs made in previous papers, and the analyses carried out in this paperis presented.

II. Background

Conceptually, the stages in distributed access control are performed in two steps: authentication and authorization. Some assumptions are necessary to model this scenario. It will be assumed that in typical distributed environments, authentication should be handled as a separate service at the clients' security domain and authorization at the service provider's domain. A typical access control flow is shown in figure 1, demonstrating how an application specific access filter, say Policy Enforcement Point (PEP), can interact in support of authentication and authorization processes [3]. The access filter in this sense can be seen as a service request filter, which determines which resources need access control, whether authentication and/or authorization are required before access is granted or denied. In order to avoid unnecessary authentication and/or authorization, the access filter can use subcomponents such as AuthnProxy (Authentication Session management) and AuthzProxy (Authorization Session Management) respectively as shown in the diagram to ensure that authenticated and/or authorized requests are not asked to perform these operations again in the same session.

Figure 1 shows that authentication and authorization services are operated in different autonomous security domains, entailing requirements for trust establishment that may inherently be complicated. In order to ensure privacy protection, the initiating client may be unwilling to supply to the PEP all the required subject attributes before service invocation. Additionally, the authentication phase validates the client's origin,

providing the PEP some attribute information that is passed to the Policy Decision Point (PDP); meaning that to complete the access decision operation, the PDP must have access to the remainder of subject's attributes to complete the decision process. But since the PDP makes the request through its Context Handler to a external entity, trust must be established. This additional request for attribute information from an external entity triggers privacy because the external entity must ensure that the requestor will treat the attribute information with some regard to privacy. Figure 1 clearly shows the complete data flow from the client initiating a service request to filtering of the request by both the authentication and authorization services when required. The next section attempts to put the developed understanding into a distributed XACML interactions context.'

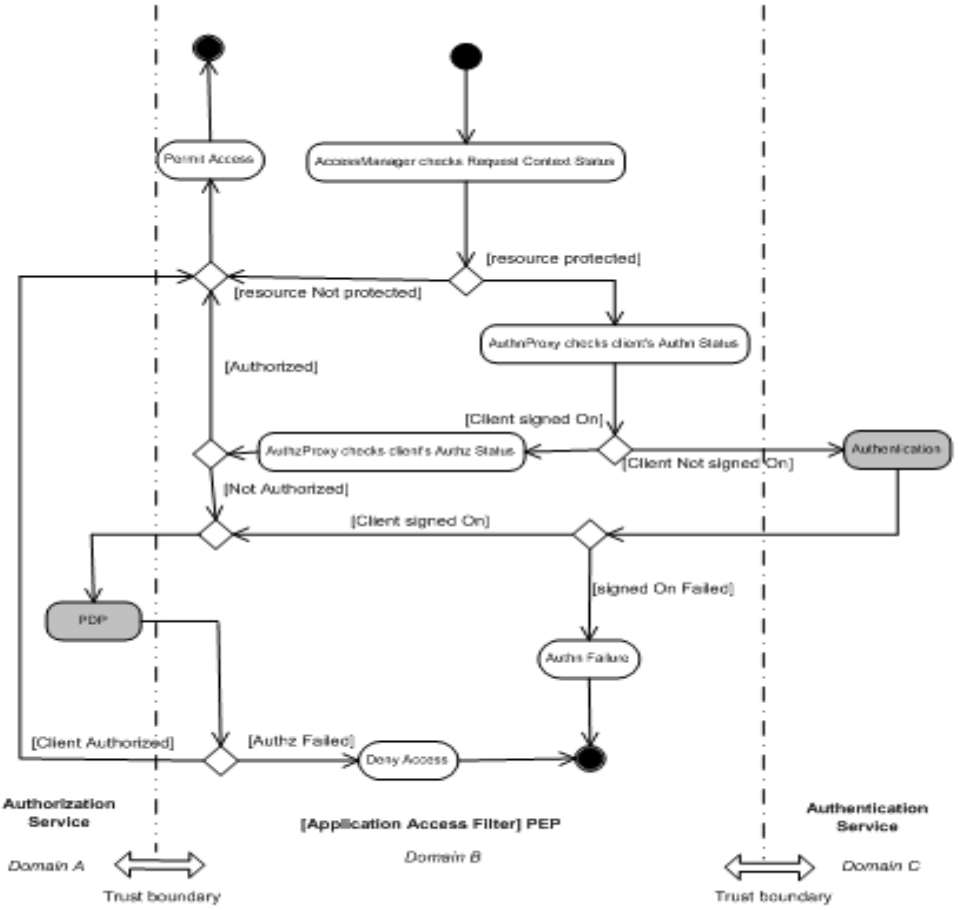

Figure 1: Conceptual Access Control Flow

III. Trust Context

In the section that follows, trust establishment is examined to further deconstruct and clarify the trust boundaries. Trust is fundamental, in distributed transactions, trust can be established between people and people, people and services, and services and services, and may demand handling the trust relationships dynamically [48]. Figure 2 depicts a simple dialogue between two parties: Alice and Bob in a typical trust context and is described as follows:

- Alice approached Bob's door and knocked;
- Bob asked 'who are you';
- Alice responded 'I am Alice;
- Bob said come in, because he recognised that was the voice of Alice and expected to see her.

Analyzing the above scenario, there is a high probability that Bob may open the door and see instead of Alice, an impostor, who mimicked Alice's voice. This naïve example can be the basis to justify the two variables associated with trust namely: behaviour and expectations. The illustration implies an innate issue of risk elements in the general concept of trust; buttressing the need to build systems that will allow communicating parties to negotiate trust based more on other properties than the PKI relationships can provide.

In the digital world, trust models are the basis of verifying and validating trust relationships, claims, privileges, properties, identity-information, etc. giving the relying party the choice of whether to trust a providing party or not based on certain defined rule constraints. Traditionally, the certainty to trust an entity can be based on the characteristics of the trust model in place.

3.1.1 Direct vs. Indirect Trust

There are two basic trust relationships: direct and indirect (sometimes referred to as transitive trust). In a typical access control operation, the service gatekeeper needs to verify and validate the claims made by a client; these operations require some sort of trust relationship. Direct trust is based on shared knowledge or a shared secret, such as username/password pair, PKI certificate, etc. which is usually established out-of-band between parties prior to communication interaction. In the case of indirect trust, a party needs

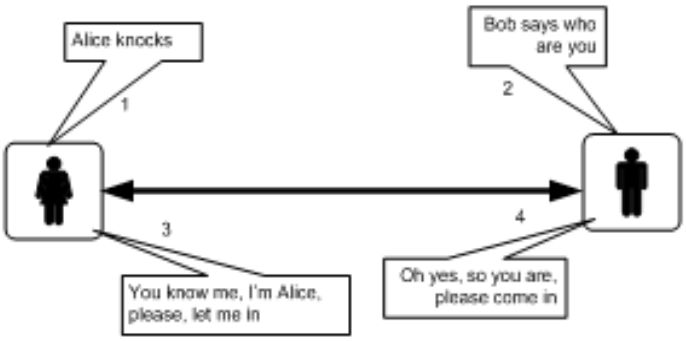

Figure 2: Typical Trust Dialogue

To validate the claims made by another party but there is no direct trust relationship between them. In other words, the parties require a trust broker, or trusted 3rd party that both of them can trust, in order to communicate and share sensitive information. Figure 3 depicts two basic primary trust models showing direct and indirect trust relationships.

Figure 3 (i) depicts the idea of direct trust between a service client (SC) and a service provider (SP) that requires a proof-of-possession to establish trust. In contrast, in indirect trust models, a 3rd party has to vouch for the identity or attribute claims, privileges, properties etc. of a party. This is illustrated in figure3 (ii).

In figure 4 (i), a more sophisticated trust model is depicted showing boundaries of security domains, as well as some kinds of possible trust relationship. Figure 4(ii) demonstrates the steps and interactions that a SC can use to request a service from an SP, when they do not have direct trust relationships. In this mode, the assumption is that direct trust between the providing party and relying party is impractical, so an intermediary, trusted by both parties must intervene.

1. The SC authenticates and obtains a signed token from its local Identity Provider (IdP) /Security Token Service (STS)

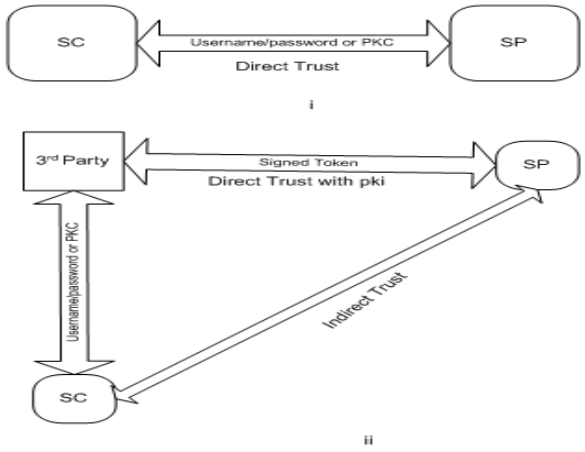

Figure 3: Basic Trust Model

2. The SC initializes and constructs another authentication request using the token from step 1 to an IdP/STS trusted by the SP. The SC's IdP/STS and SP's IdP/STS need to have prior trust relationships for this protocol to succeed. The SP's IdP/STS validates and processes the token request, issues a fresh token or cross certifies the token issued by the SC's IdP/STS, and returns it to the SC.
3. The SC constructs a web services message with the token and requests a service from the SP. The SP validates and processes the request and sends the appropriate response to the SC.

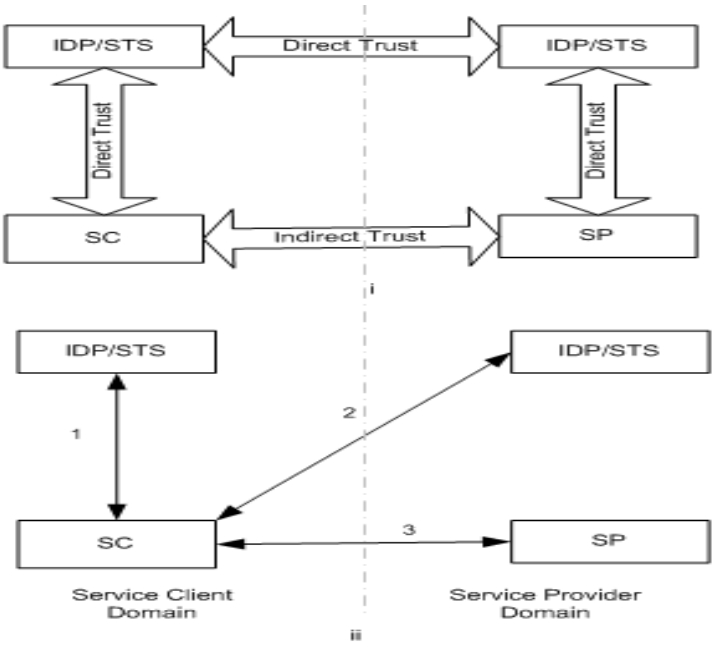

Figure4: Brokering of Trust via Two TTPs

It can be seen that the basis of trust between the SC and the SP can be analyzed as follows. The direct trust between the SC and its IdP/STS exists in the form of a username/password pair. In contrast, the SC has no direct trust with either the SP or its IdP/STS. Thus, trust between the SP and SC, which has no direct path, is provided by the trust between the two IdP/STS respectively. It can be seen that from the SP's viewpoint, the SC's IdP/STS has no direct trust relationship with it, but it can trust the assertion claims based on its relationship with its IdP/STS. This trust model exposes some inherent security flaws, which can be exploited by a greedy participant in high risk business transactions. The recipient party merely trusts the assertion claims made by a 3rd party on the identity of the holding party, which are unlikely to reveal all the intentions of that party. Arguably, this trust establishment cannot solely be relied upon to create confidence in an entity, but can provide the foundation upon which parties can negotiate and build more trust, through the mutual exchange of what they expect from the other party, and what they are willing and able to do for that party.

On the basis of the above understanding, it can be concluded that in distributed environments where multiple actors exist in multiple domains, it is essential that brokered trust establishment be utilized as an 'introductory trust', upon which a higher threshold of trust can be built by the parties themselves. Having said this, it is equally vital to point out that what will determine the trust ascribed to a party might be based on some measurable risk metrics, which are outside the scope of the present work. To sum up, it can be concluded that privacy assurance cannot sufficiently be based on the provision of PKI relationships only, but requires more dynamic exchange of obligating and binding constraints between the parties themselves as the basis for building a higher level of privacy trust. Nevertheless, PKI still has an important role to play.

IV. Acme in a Distributed Context

The XACML confined its scope to access control model and language constructs, and intentionally overlooked how XACML actors in distributed environments can collaborate in a mutual access control interactions. The assumption is that it can utilize other complementary standard models to describe assertions, protocols or transport mechanisms that will enable its distributed actors to converse in secure trusted manner. In practice, the PEP entity is responsible for protecting access to the resources. The PEP filters every access request and applies appropriate enforcement, which may include establishing the authenticity of the request or sending the description of the request to the PDP entity in the form of XACML context request. Two things can

be established here. First, in the case of authenticity, it may require a formal authentication of the initiator by any available means. Second, the PDP needs to evaluate the request against its available policies and attributes to make an authorization decision, which has to be passed back to the PEP.

Traditionally, XACML PEP obtains a description of the request context from a range of possibilities including distributed (perhaps on-line) *Attribute Authorities* (AA) or *Attribute Repositories* (AR) for the subject designator attributes. Similarly, the PDP may interact in distributed manner to obtain the policies from PAP or from *Policy Repositories*. Furthermore, the PEP may not necessarily supply all the subject's attributes at the instance of making a request context from the PDP; in this case, the PDP through its Context Handler component will attempt to augment the subject attributes by asking a designated AA or AR for other attributes. This particular convenience is what makes XACML a candidate for protecting privacy and confidentiality in distributed environments. Of course, this raises the issue of trust; the distributed components must trust each other. Where the components exist in autonomous security domains, the level of trust required may vary, and even become complex. It is therefore important to examine how XACML actors can interact in distributed environments which will help in understanding the rest of the sections.

4.1.1 Distributed XACML Context Interaction

Figure 5 shows XACML actors in a distributed environment and their interactions which are described below: (It is assumed that the client has successfully authenticated with its local authentication service and token(s) issued in that respect in the form of assertion)

1. The client constructs a service request with the token containing the information about the subject and sends it to the PEP; it is important to note that because of privacy concerns, not all the subject information is included in this phase.
2. The PEP constructs a request context containing the token as the subject descriptor, and obtains other information, i.e. properties of the resources, time constraint, etc, and presents it to the Context Handler;
3. The Context Handler formats the request context appropriately and presents it to the PDP to decide whether access should be allowed;
4. The PDP obtains all applicable policies and evaluates them against the request context. If the PDP cannot complete its operation because of missing subject attributes, it requests the missing attributes from the Context Handler;
5. In turn, the Context Handler requests the missing attributes from the client's PIP (It is assumed that the client performed authentication with its PIP before service invocation). The dotted line indicates this external conversation;
6. The client's PIP returns the missing client's attributes to the Context Handler;
7. The Context Handler sends the returned attribute information on to the PDP;
8. The PDP completes the evaluation process and passes decision results to the Context Handler;
9. The Context Handler formats the decision results into a response context and sends it to the PEP;
10. The PEP interprets the response context and enforces a decision by either allowing the requested resources or indicating that access is denied to the client.

Analyzing the above interactions, it becomes apparent that the steps in the dotted lines require some form of trust to be established, to ensure privacy and confidentiality. In the scenario where the client has to reveal more attribute-information to a remote party before the access control decision is taken, privacy becomes a serious issue. The interactions above involve communication between external entities in different security domains, increasing the need for privacy, confidentiality and trust in distributed environments. This implies that for adequate privacy protection, negotiations between the external entities are desirable. This is necessary to allow both parties to determine how, where and when to reveal resources and attribute information, and apply desirable obligating constraints on the other party to guarantee privacy and confidentiality. In this regard, it is important to mention that trust is the vehicle for achieving these goals. Based on the above assessment and knowledge, the following necessary assumptions are made[1]:

[1] Authentication is not the primary focus of this work as most of the existing authentication approaches can be integrated with the described framework.

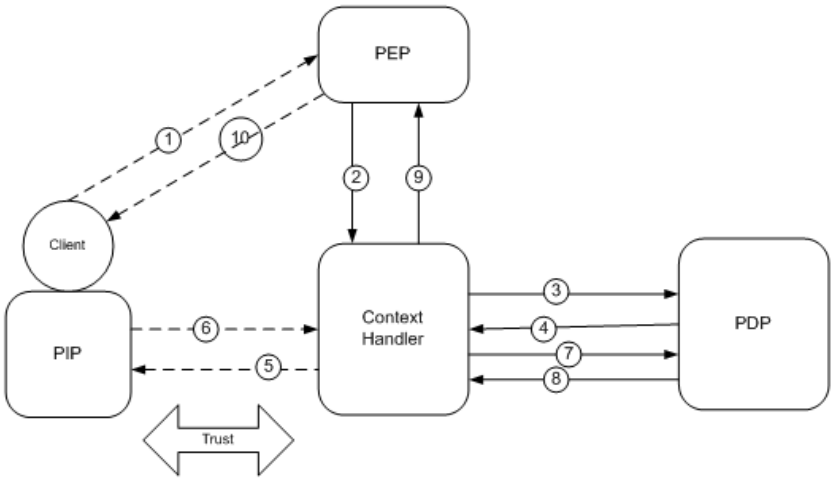

Figure 5:XACML Components Interaction Context

- No access is allowed to a protected resource by default. The access rules for a protected resource must require attribute information of the client making the request before access is allowed, otherwise access is denied. Where more than one attribute is required by the access control operation, the rules shall be expressed in a manner such that one of the subject descriptors provides the initial mechanism to establish the first degree of trust context.
- The client may not be willing to disclosure applicable attributes of privacy concern at the first stage of the access request. In this regard, the authentication phase between the client and its local authentication provider, in which a token is issued, partially reveals information (i.e. origin authenticity) about the client, to serve the above-mentioned first degree of trust context.
- The initial information provided by the client is not sufficient to breach the client's privacy or the confidentiality of the protected resources. This potentially defeats any attempt by a bogus participant to mount probing attacks, often associated with trust negotiation [10]. The underlying theoretical assumption is that if the interacting parties decide to withdraw from the transaction at this stage, they are not overtly exposed to privacy and confidentiality risks. Moreover, if any of the participants is an imposter, then the initial access rule filter will screen out the request, and the imposter will not succeed in any subsequent interactions. Although it can be argued that having a clue about the origin of a party is a privacy risk, what is important is the privacy risk impact factor and its actual consequences, which are outside the scope of this thesis.
- Where the first degree of trust establishment described above is insufficient to gain access to protected resources, both parties require other levels of trust establishment to reach their various goals. In this scenario, any attempt by the service to request more of the client's attributes will trigger mutual trust negotiations before sensitive information is exchanged.
- To make this negotiation phase privacy compatible, the service simply sends its access control policy as requirements across to the client. The client, uncertain whether the service will respect its security preferences, cannot reveal sensitive information, but can respond with a similar counter policy. This iterative process triggers privacy trust negotiation and exchange of relevant attribute information, which can take a number of rounds until both parties are satisfied to release their various sensitive resources.
- It is assumed that the above scenarios do not guarantee assurance that the parties will respect each other's privacy, so additional steps are needed; this prompted a strong consideration of a workable protocol that will enable communicating parties to generate and exchange difficult-to-repudiate tenable evidence about their contextual information, in order to provide end-to-end privacy and confidentiality.

To further the above suppositions and substantiate them in the proper context, a security threat modelling technique is utilized to critically survey and scope the privacy problems in an application environment [89]. The benefit of security modelling is to ascertain the extent of security to apply in a given application domain, through a proper analysis of inherent and foreseeable security vulnerabilities and threats, and determine suitable mitigations. In the next section, this approach is used to investigate the casual effects of privacy and confidentiality in an access control environmental context in an attempt to validate the earlier assumptions made about privacy and confidentiality.

V. Conclusion

This paper describedthe conceptualization of privacy in access control in distributed environment context. The analysis of the XACML actors in a distributed environment, are critical to the understanding of how to use XACML to address the privacy problem. Equally, the investigation and understanding of the trust models are instrumental in determining how trust relationships can improve privacy assurance in distributed environments. Overall the rationale for the choice of XACML and WS-XACML as policy candidates for access control frameworkin distributed context resulted from the various analyses of the requirements for addressing privacy and confidentiality problems considered.

References

[1]. R. Vidgen, D. Avison, B. Wood, and T. Wood-Harper, Developing Web Information Systems: Butterworth-heinemann Information, 2003.
[2]. B. Carminati, E. Ferrari, and H. Patrick C.K, "Exploring Privacy Issues in Web Services Discovery Agencies," IEEE Security and Privacy, vol. 3, pp. 14-21, 2005.
[3]. M.Lorch, S.Proctor, R.Lepro, D.Kafura, and S.Shah, " First Experience Using XACML for Access Control in Distributed Systems," presented at ACM Workshop on XML Security, Fairfax Va US,, 2003.
[4]. E. Bertino, E.Ferrari, and A. Squicciarini, "Trust Negotiations: Concepts, Systems and Languages," IEEE Computer, pp. 27-34, 2004.
[5]. N. Huntley, "Open and Closed Systems," http://www.users.globalnet.co.uk/~noelh/OpenClosedSystems.htm 2003.
[6]. A.Acquisti, "Privacy and Security of Personal Information- Economics Incentives and Technological Solutions," presented at Workshop on Economics and Information Security, University of California Berkeley, 2002.

IOSR Journal of Computer Engineering (IOSR-JCE) is UGC approved Journal with Sl. No. 5019, Journal no. 49102.

U. M. Mbanaso "Privacy in Distributed Access Control Environment." IOSR Journal of Computer Engineering (IOSR-JCE) 20.3 (2018): 27-33.