XPACML
eXtensible Privacy Access Control Markup Language

Kheira Bekara
Institut TELECOM
Telecom SudParis
CNRS Samovar UMR 5157
Evry
France
Email: Kheira.Bekara@it-sudparis.eu

Yosra Ben Mustapha
Institut TELECOM
Telecom SudParis
CNRS Samovar UMR 5157
Evry
France
Email: yosra.ben_mustapha@it-sudparis.eu

Maryline Laurent
Institut TELECOM
Telecom SudParis
CNRS Samovar UMR 5157
Evry
France
Email: Maryline.Laurent@it-sudparis.eu

Abstract—**Privacy in the digital world is a critical problem which is becoming even more imperious with the growth of the Internet, accompanied by the proliferation of e-services (e.g. e-commerce, e-health). One research track for efficient privacy management is to make use of user's and service provider's (SP) privacy policies, and to perform an automatic comparison in between to help any (skilled or unskilled) users preserving their privacy.**

In this paper, we focus on the privacy policy comparison issues. We adopt the eXtensible Access Control Markup Language (XACML) as a policy description language for user's preferences and SP's policies. We enrich XACML with P3P main elements to permit a privacy aware access control on the user's personal data elements, thus resulting in the new XPACML (eXtensible Privacy Access Control Markup Language) language.

The paper describes first the XPACML language. Then, it presents the functional architecture at the user's side where the automatic privacy policy compliance can be performed. Finally it discusses our contributions compared to the main proposed solutions in the literature to better identify the interest of them.

I. Introduction

The changes brought by concepts, such as e-Government and e-Commerce, are moving internet users towards electronic based services whatever public or private usage. These services require more and more user's personal data to facilitate interactions between citizens and organizations, and personalize services for user's convenience.

Unfortunately, in Internet, there are a number of situations where Privacy loss can occur [1] [2] [3]u. A noticeable number of identity thefts, loss reports and misuses of personal data have been declared. For instance, in 2008 about 9.9 million Americans were reported as victims of identity thefts, with an increase of 22% compared to 2007 [4].

Personal data are being considered of high interest target for service providers (SP). Moreover, advanced data processing techniques such as sophisticated databases, data mining, profiling techniques... enable SPs to study the web user's behavior, and to produce an accurate profile, even from anonymous data. As a consequence, web users are becoming increasingly concerned about their privacy. About 72% of Internet users give up their online purchases when they are requested to provide personal data. Thus, e-Privacy is a critical issue that requires urgent and high investment by scientists to make users feel trusting in e-services.

On the one hand, traditional security safeguards do not ensure privacy protection [5]. On the other hand, there is no tool allowing neither the SP to express an understandable privacy policy, nor the user to express his preferences regarding the use of his private data.

E-Privacy aims to protect users' personal data, and particularly to permit web users to get control on their personal information. E-Privacy is defined in [6] as the ”ability of an individual or group of individuals to stop information about them from becoming known to people others than those they choose to give the information . E-Privacy is related to confidentiality, secrecy, anonymity... but it requires more than that. It must answer to the following questions:

1) Who is collecting my data?
2) Why is he requesting them?
3) How long will he keep them?
4) Will he share them? if so, with whom?

To meet this need, several specifications have been defined and adopted to technically protect personal data.

The policy-based e-Privacy management is a hot research topic, especially privacy policy negotiation protocols. The first important step in such process is the definition of a suitable policy language. Many of them were defined. W3C developed P3P specifications and APPEL as a complementary language to express respectively the SP's privacy policy and user's preferences. Unfortunately, these languages do not support privacy policy negotiation.

In this paper, we propose to define a privacy policy language based on the efficient access control language: XACML (eXtensible Access Control Markup Language). That is, we consider the e-Privacy problem as a problem of access control to user's personal data. New elements are defined in the

XACML Policy Model Language, like P3P basic tags, and Service Type. This latter is introduced based on the idea that privacy is context dependent. This is true since the user's preferences are adapted to the service type they are interacting with.

The main objective of these works is to establish a trust relation between the SP and the web user. The proposed policy language is an essential step for the SP and users to understand each other's privacy requirements. This is a first step towards defining an automatic negotiation protocol.

The remainder of this paper is organized as follows. Section 2 briefly describes some of the existing privacy description languages. Section 3 introduces our XACML approach and section 4 presents the framework at the user side. Section 5 compares our approach to existing privacy-aware languages. Section 6 leads with pertinent conclusions and perspectives.

II. Existing Privacy Policy Languages

A. The P3P Policy Language

W3C developed the P3P specification to enable web sites expressing transparently their privacy policy in a standard machine-readable format [7]. These policies are processed automatically by enabled P3P web browsers during online transactions. The main contribution of P3P can be summarized by the following two points:

1) P3P expresses traditional privacy policy in a computer-readable format as it uses standard XML tags;
2) P3P defines the vocabulary set for each element in use.

A P3P policy contains basically a number of statements that aggregate several data-groups. Each data-group describes the data elements and the corresponding policy elements that apply to. The major XML elements that P3P includes in a statement are listed below:

- DATA: This tag expresses the data item collected by the SP. Data items are grouped into Categories. In a P3P policy, data elements having the same policy can be grouped into Data-Group element.
- PURPOSE: It expresses why the SP is requesting data;
- RECIPIENT: This tag contains the list of other SPs that will share the collected data;
- RETENTION: It indicates the period of time during which the collected data will remain stored in the SP's DB (DataBase).

B. Expressing user's preferences with APPEL and XPref

The P3P Privacy Policy Exchange Language APPEL [8] was defined by W3C to complement the P3P language, and to compare the P3P privacy policy against the user's preferences. APPEL is used to express the user's preferences on a machine-readable format since it is based on the XML dialect.

APPEL was initially designed with limited scope, but it truly offers an attractive and simple schema to express the user's preferences. Unfortunately, [9], [10] and [11] show that this language contains serious drawbacks. Beyond the limits listed in the APPEL specification [8] like the uncapability to express sophisticated rules, [9] demonstrates a design error of APPEL. Four major deficiencies can be identified:

- What is acceptable can not be easily specified;
- A matching policy might be rejected;
- P3P extensions are not supported so they are by default permitted;
- Simple combinations are hard to express.

These drawbacks were mainly due to the use of connectives and the limited interoperability with P3P. To overcome these APPEL deficiencies, researchers [9] proposed the XPref language. XPref reuses two XML elements of APPEL (RULESET and RULE) and makes use of XPath to specify acceptable or unacceptable combinations of P3P elements. However, as XPref is designed only to define the user preferences, and our objective is performing negotiation between privacy policy and preferences, we rejected the XPref language.

C. Using XACML for a Privacy Language

XACML (eXtensible Access Control Markup Language) is first an Access Control reference model developed by OASIS, [12]. XACML is not only an Access Control Policy language but also an Access Control request/response language. Several works study the capabilities of XACML to support policy compliance.

Moreover, all the messages exchanged in the XACML architecture and XACML policies are written in XML dialect. XACML stores the policy objects in a hierarchy of policy sets, policies, and rules. An XACML Policy is made up of Rules and a Target. Policies are grouped into PolicySet elements. As such, XACML is providing a mechanism that offers an advanced high-level Access Control. It consists in a finer granular access control than simply denying or granting access. XACML specification just deals with the framework. It has various extensible points that we can adopt. It can operate in different environments, depending on how developers implement and use the policy points defined in XACML specification. This explains why XACML is listed among existing Privacy Preserving Languages. In [6], [13], [14], [15]..., the authors conclude that preserving privacy using XACML seems to be an interesting solution to define both the user's preferences and the SP's privacy policy. However, some changes and extensions to the basic specifications are necessary.

D. The Enterprise Privacy Authorization Language: EPAL

Contrary to XACML, EPAL is a language which is designed for privacy protection. It was defined by IBM, and approved by W3C. EPAL supports policy exchange respecting a structured format. For each data, EPAL specifies a context and an obligation. The latter claims the processing that is done by the Enterprise Information System. EPAL is used to communicate the Enterprise's Policy to users, and to apply internally this policy thanks to compatible systems.

Like XACML, EPAL may be used to enforce control over data, but with a less precisely defined architecture, EPAL requires support from third applications to perform Data Access

Control. With similar XACML structures, EPAL is considered as a subset of XACML dedicated to privacy protection, but the interest of it is limited to standardizing the collected data vocabulary. There are some additional differences between those two languages. For instance, several functions related to access control granularity are available in XACML but not in EPAL, like: unauthorizing access to part of a hierarchical resource, unauthorizing a subject to play multiple roles at the same time

Hence, EPAL's interest is limited, especially as XAMCL is now widely deployed and widespread.

III. Our XPACML Approach

A. The Policy Structure

As mentioned in the introduction, privacy protection level plays an essential role in user acceptance and trust in any e-Service. Access control is one of the most promising research field to support privacy at both user and server sides.

XACML OASIS standard is one of the most widely used access control languages. It is considered as a standard policy language for web services. It serves to describe policies and to support access control decisions. The scope of application domains is very large. XACML is designed to support centralized or decentralized policy management and has been widely deployed.

In our approach, we adopted a policy based privacy management approach, and thus we naturally selected XACML as a language to express privacy aware access control policies. Our approach consists in defining an innovative privacy acces control-based system. It includes a novel privacy policy structure and privacy user side architecture with the objective to help designing an automatic privacy preserving tool. By considering user's personal data as resources, we define new XML elements in the Target tag of XACML. As shown in figure 2, we introduce P3P main tags: ”purpose, recipient, retention” in the Resource element of the XACML policy model language. This enables the adoption of the main privacy vocabulary defined by P3P platform, and their integration into a policy access control model. Moreover, each Resource element is identified by a unique identifier: ResourceId.

An additional element Service_Type is also defined. It describes the service category of the SP for which the policy applies. This element enables both the SP and the user to define respectively their policies and preferences for a given Service Type. This avoids the PolicySet element containing policies for each SP. For the moment, we consider six Service_Type values: e-Commerce, e-Government, e-Banking, e-Telecom, e-Health, e-Learning.

The advantage of these changes is twofold. First, it allows both SP and user to express respectively their privacy policy and privacy preferences using the same policy structure and vocabulary. They should differ only by the Effect attribute in the Rule element. This attribute is not present in the SP privacy policy since this latter is implicitly a request. Second, the user can define his preferences regarding a specific Service Type.

```
<?xml version=”1.0” encoding=”UTF–8”?>
<xacml:PolicySet ... >
  <xacml:Description>User A’s Privacy Preferences </
        xacml:Description>
  <xacml:Policy ...>
  <xacml:Description>this is the privacy preferences for
        the e–commerce service </xacml:Description>
  <xacml:Target>
    <xacml:Subjects>
      <xacml:Subject>
        <xacml:SubjectMatch ...>
          <xacml:AttributeValue .../>
          <xacml:SubjectAttributeDesignator .../>
        </xacml:SubjectMatch>
        <xacml:Service_Type>e–Commerce</xacml:Service_Type>
      </xacml:Subject>
    </xacml:Subjects>
  </xacml:Target>
  <xacml:Rule Effect=”Permit” RuleId=”RuId”>
  <xacml:Description/>
  <xacml:Target>
    <xacml:Resources>
      <xacml:Resource ResourceId=”ReId”>
        <xacml:ResourceMatch ...>
          <xacml:AttributeValue ...> address </
                xacml:AttributeValue>
        </xacml:ResourceMatch>
        <p3p:PURPOSE>
          <p3p:current required=”always”/>
        </p3p:PURPOSE>
        <p3p:RECIPIENT>
          <p3p:ours/>
        </p3p:RECIPIENT>
        <p3p:RETENTION>
          <p3p:no−retention/>
        </p3p:RETENTION>
      </xacml:Resource>
    </xacml:Resources>
    <xacml:Actions>
      <xacml:Action>
        <xacml:ActionMatch ...>
          <xacml:AttributeActionValue>read </
                xacml:AttributeActionValue>
        </xacml:ActionMatch>
      </xacml:Action>
    </xacml:Actions>
  </xacml:Target>
</xacml:Rule>
</xacml:Policy>
</xacml:PolicySet>
```

Fig. 1. The User's XPACML policy preference example.

B. The Policy Structure Extensions

From a structural point of view, the main components of the XACML Policy structure are ”Policy Set”, ”Policy”, and ”Rule”. The main changes introduced by our approach are within the Target unit of the Policy and Policy Set elements. As defined by the OASIS standard, the Target unit defines a set of: Resources, Subjects, Actions, and Environments. The Subject, within a Target element, describes the SPs requiring personal Data-Elements. The Rule is intended to apply to all SPs of the same type, and on the Data-Elements identified under the ResourceId. Hence, the P3P basic tags (purpose, recipient, retention) are introduced as components of the Resource element. They permit the expression of the privacy policies like attributes of the Resource element using P3P vocabulary.

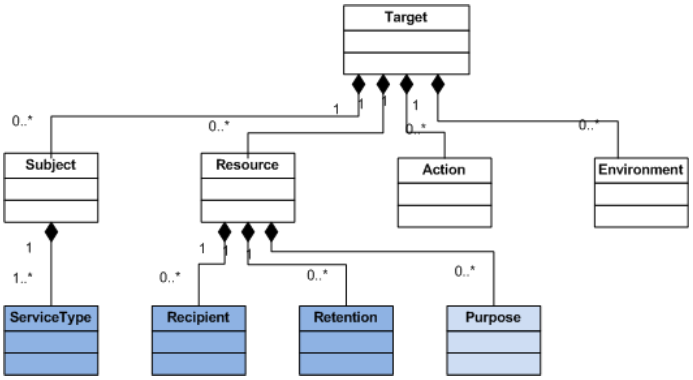

Fig. 2. Proposed elements.

The actions permitted on a data element (a Resource) are expressed by the Action element. It enables defining the usage intended by the SP. We define three types of actions:

- read: when the SP is requesting data only to achieve the current purpose.
- collect: when the SP wants to store collected personal data.
- share: when the collected personal data will be shared with listed Recipients (third parties).

The Environment element will be used in the future to express the transaction context. This element is out of scope of this paper.

IV. The XPACML Functional Framework for Privacy Support At User's Side

To allow the Privacy policy negotiation, we bring some modifications to the XACML structure. Figure 3 contains the main components defined in our approach.

- Privacy Administration Point (PAP): This component interacts with the user, and permits to define the user's privacy preferences.
- Privacy Policy Decision Point (PPDP): This component compares the SP's privacy policy and the user's privacy preferences based on the XACML structure presented in the figure. 1
- Data Information Point (DIP): This component is activated by the PPDP only when privacy policy and preferences are matching.
- Privacy User Agent (PUA): It is a complementary module for the browser. It captures the SP's privacy policy from the http stream at the beginning of the transaction. Then this agent redirects the captured policy to the PPDP. Also, it redirects the PPDP's decision to the corresponding SP.

Hereafter the data flows at the user's side are described:

1) The PUA captures the SP privacy policy which is then redirected to the PPDP;
2) The PPDP identifies the service type of the SP. Then, it requests for the corresponding user's preferences;
3) The PPAP sends the requested preferences to the PPDP;
4) The PPDP then compares the SP privacy policy against the user's preferences;

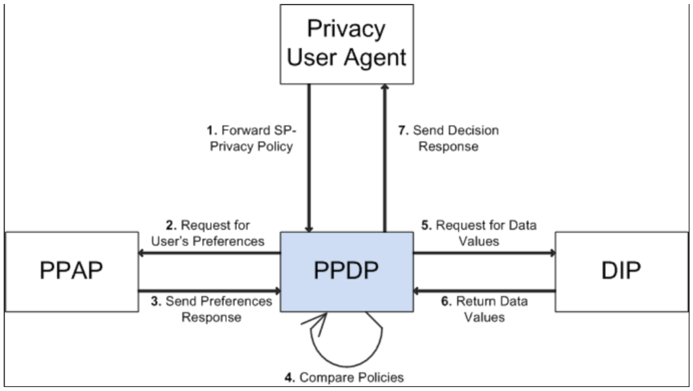
Privacy User Agent
1. Forward SP-Privacy Policy
7. Send Decision Response
2. Request for User's Preferences
5. Request for Data Values
PPAP
PPDP
DIP
3. Send Preferences Response
6. Return Data Values
4. Compare Policies

Fig. 3. Data Flow Diagram on the User Side.

5) This step occurs only when the SP's policy matches the user's preferences. The PPDP requests the attribute values from the PIP;
6) The DIP sends the corresponding values to the PPDP;
7) Finally, the PPDP sends the privacy decision and possibly the attribute values in case the decision is accept access.

V. Analysis

A lot of efforts have been devoted to the privacy enhancing mechanisms based on policy management. The work done by IBM [16] permitted organizations to express their privacy policies using the P3P vocabulary. At the user's side, the implementations proposed by AT&T [17] were the first integrating the P3P vocabulary into user agents. Implementing privacy negotiation process during the transaction between the SP and the user can overcome four major shortcomings of the above cited works:

- The ”Take it or leave it” principle (the user can only accept or deny the SP's proposal as a whole.
- The ”One size-fit-all” principle (the same privacy policy is proposed to all interested users).
- The global expression of the privacy preferences: in [privacy bird], three levels are defined for privacy preferences (hight, medium, low) for groups of data elements (financial data elements, health data elements, ...).
- At the implementation level, the comparison is based on the P3P cookies (not on the P3P privacy policy itself).

The consideration of the privacy policy in the fine granular level has not been investigated by these works, and only the policy comparison based on a group of data elements or policies has been considered.

We can conclude that the privacy is not yet considered in a fine granular level. Hence, the negotiation step still remains theoretical. Also, to the best of our knowledge, there is no implementation of the comparison of privacy policies between a SP and a user, taking into account a fine grained data element level and the Service Type category. Our proposed solution overcomes these deficits:

- It offers to the user the possibility to define his privacy preferences at a high level, or a fine grained level for each data element (detailed expression of the policy).
- It defines the same privacy policy structure both for the User and Server Sides, and hence permits a negotiation adapted to the needs of either the user or the SP.
- It fixes the Service Type as an enter point to facilitate the comparison of the SP's privacy policies and the user preferences.

VI. Conclusion

This paper describes a new approach for protecting the user"s privacy in an electronic transaction using XACML and P3P.

Our approach combines the advantage of research works done in the privacy management and the policy-based control access. We define a novel privacy aware XPACML Policy Language Model. The architecture of the privacy policy compliance and the data flow between the user and the server side are also presented. The main idea of this approach is the adaptation of the XACML framework and the integration of the P3P main elements. Our solution permits both users and SP to define their privacy preferences and policies in a common XACML-compatible format. Our solution can help comparing the preferences and the policies based on the XACML framework.

We plan to extend our work along the following directions. The first direction is to explore the comparison of privacy policy and preferences based on the proposed policy structure. Also, the use of an ontology to express privacy policies, and ontological reasoning need to be investigated. Our goal is to enhance the privacy policy negotiation process using XACML and semantic reasoning techniques.

Acknowledgements

This research is part of the project called FC2 (Federation of Circles of Trust www.fc2consortium.org). Authors are thankful to DGCIS (Direction gnrale de la comptitivit de l'industrie et des services) for financially supporting TELECOM SudParis.

References

[1] P. M. Schwartz, *Internet Privacy and the State*. Connecticut L, 2000.

[2] F. H. Gate, *Principles of Internet Privacy*. Connecticut L, 2000.

[3] P. Trudel, *Reinventing Data Protection*. Springer, 2009.

[4] K. M. Finklea, "Identity theft: Trends and issues," *Congressional Research Service*, January 2010.

[5] J. Meyer, "How to manage, negotiate, and transfer personal information on the Web," Ph.D. dissertation, University of Applied Sciences of Hamburg, 1999.

[6] D. K. W. C. Vivying S. Y. Cheng, Patrik C. K. Hung, "Enabling web services policy negotiation with privacy using xacml," in *IEEE 40th Hawaii International Conference on System Sciences*, 2007.

[7] "P3P: Platform for Privacy Preferences," http://www.w3.org/TR/P3P11/.

[8] "APPEL 1.0: A P3P Preference Exchange Language 1.0," http://www.w3.org/TR/P3P-preferences/.

[9] "XPref: a preference language for P3P," http://www.sciencedirect.com/.

[10] Y. B. Layth Sliman, Frdrique Biennier, "A Security policy framework for context-aware and user preferences in e-services," *Journal of System Architecture 55*, 2009.

[11] G. Hogben, "A technical analysis of problems with P3P 1.0 and possible solutions," *W3C Workshop on the Future of P3P*, November 2002.

[12] "eXtensible Access Control Markup Language (XACML) version 3.0," http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml.

[13] H.-Y. G. Yuh-Jong Hu and G.-D. Lin, "Semantic Enforcement of Privacy Protection Policies via the Combination of Ontologies and Rules," *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing*, 2008.

[14] A. Anderson, *Web Services Profile of XACML (WS-XACML) Version 1.0*, August 2007.

[15] M. Y. B. Laurent Bussard, "Can Access Control be Extended to Deal with Data Handling in Privacy Scenarios?" November 2009, http://research.microsoft.com/apps/pubs/default.aspx?id=105065.

[16] "IBM P3P Policy Editor," http://www.alphaworks.ibm.com/tech/p3peditor.

[17] "Privacy Bird," http://www.privacybird.org/.