Extending XACML Access Control Architecture for Allowing Preference-Based Authorisation

Gina Kounga, Marco Casassa Mont and Pete Bramhall

HP Laboratories
HPL- 2009-361

Keyword(s):
Privacy, Access controls

Abstract:
European data protection regulation states that organisations must have data subjects' consent to use their personally identifiable information (PII) for a variety of purposes. Solutions have been proposed which generally handle consent in a coarse-grained way, by means of opt in/out choices. However, we believe that consent's representation should be extended to allow data subjects to express a rich set of conditions under which their PII can be used. In this paper we introduce and discuss an approach enabling the representation of consent as fine-grained preferences. To enforce such consent, we leverage and extend the current standard XACML architecture and framework. As data collectors maintain links between PII and associated preferences, preferences should also be considered as part of this PII. Therefore our solution prevents access control components from directly accessing any PII.

External Posting Date: November 21, 2009 [Fulltext] Approved for External Publication
Internal Posting Date: November 21, 2009 [Fulltext]

Published in the proceedings of the 7th International Conference, TrustBus 2010, Bilbao, Spain, August 30-31, 2010.

Extending XACML Access Control Architecture for Allowing Preference-Based Authorisation

Gina Kounga, Marco Casassa Mont and Pete Bramhall

Hewlett-Packard Laboratories
Long Down Avenue
Stoke Gifford
Bristol
BS34 8QZ
United Kingdom
{Gina.Kounga, Marco.Casassa-Mont, Pete.Bramhall}@hp.com

***Abstract*. European data protection regulation states that organisations must have data subjects' consent to use their personally identifiable information (PII) for a variety of purposes. Solutions have been proposed which generally handle consent in a coarse-grained way, by means of opt in/out choices. However, we believe that consent's representation should be extended to allow data subjects to express a rich set of conditions under which their PII can be used. In this paper we introduce and discuss an approach enabling the representation of consent as fine-grained preferences. To enforce such consent, we leverage and extend the current standard XACML architecture and framework. As data collectors maintain links between PII and associated preferences, preferences should also be considered as part of this PII. Therefore our solution prevents access control components from directly accessing any PII.**

***Keywords*: Privacy, Access controls**

1 Introduction

Data protection regulations [1, 2] require organisations to process collected personal data only with data subjects' (e.g., end-users) consent for that processing. In the literature, this requirement is usually translated into opt in/out mechanisms that permit to capture data subjects' consent [3, 4]. However, opt in/out mechanisms do not provide any freedom to data subjects to fully specify how they would like to limit their personal data to be used. These mechanisms are indeed mainly associated with consent forms specified by data collectors (the entities that collect personal data items from data subjects) for the data subjects, which leaves data subjects only limited control of their personal data.

We believe that the notion of consent has to be extended to encompass these needs. In the context of this paper, we define consent as *a set of fine-grained privacy preferences that define the actions that can be performed on a personal data item or a group of personal data items*. The data collectors still define the preference framework, but they explicitly share the management of preferences with data subjects. Therefore, the value of the privacy preferences to be associated with each personal data item can be set by the data subject, along with the personal data items they apply to, before sending this information to the data collector. As proposed by Karjoth et al in [4], the data collector stores each personal data item as well as the corresponding preferences and maintains a link between both to guarantee that each preference can always be associated to the data item it applies to. Consequently, preferences can be linked to a living individual and therefore have to be considered as being personal data [1].

In this context, enforcing consent requires guaranteeing that each personal data item is accessed only if the conditions expressed by the associated preferences are met. As the data collector may collect thousands of data items from thousands of different data subjects, enforcing consent, as previously described, introduces a scalability problem. Further, as preferences are personal data, they also need to be securely maintained and only accessed by authorised principals – i.e., unique entities.

In this paper, we propose a solution that ensures that only the legitimate entities/data receivers can access personal data. For that, we propose an extension to the OASIS eXtensible Access Control Markup Language (XACML)

The original publication is available at http://www.springerlink.com/content/978-3-642-15151-4#section=761233&page=1&locus=0

[5] architecture and framework to enforce consent based on fine-grained preferences representing data subjects' consent. The XACML choice is influenced by the fact that this framework is currently a reference standard. Our solution builds on the observation that in most organisations, personal data are collected and managed by specific entities – e.g., the human resources service, the customer management service, etc. The manner in which these entities manage personal data is dictated by a set of regulations such as employment laws. Consequently, these organisations are constrained to use personal data as specified by these entities. In our solution, an *attribute authority* (AA) represents such an entity. The AA is the entity within the data collector which collects and stores personal data items and the associated preferences. It is composed of subcomponents that extend the XACML access control architecture to allow access control decisions to be made based on preferences' values. The goal is to ensure that no XACML component – i.e., neither the policy decision point (PDP) nor the policy enforcement point (PEP) – accesses the preferences and the personal data items. Only the AA and the authorised principals – that have been granted the access – do access them. As the AA is designed to adapt to any type of data repository or data store, our solution does not require heavy modifications to be performed on organisations' legacy systems to make it work. To the best of our knowledge, no other solution has been proposed which enforces consent based on fine-grained preferences, protects the access to these preferences and the associated personal data items and which, at the same time, can adapt to any legacy system.

This paper is organised as follows. In Section 2, we present the scenario that we consider. This will be used as a reference in the remainder of this paper. Then, in Section 3 we discuss the related work. In Section 4, we present the assumptions on which our solution relies. We present our proposed extended XACML architecture in Section 5, and detail in Section 6 the interactions allowing the enforcement of consent based on fine-grained preferences and the protection of personal data. Finally, we present the current status of our work in Section 7 and conclude our paper in Section 8.

2 Scenario

In this paper, we consider a scenario in which an organisation needs to collect some personal data from individuals in order to provide some services to these individuals. During collection, individuals specify fine-grained privacy preferences defining the conditions under which their personal data should be accessed. After being collected, these personal data need to be accessed and processed by various business processes within the organisation, in order for the relevant services be provided to individuals. To enforce consent, access to personal data by a business process is only to be granted if the conditions defined by individuals with their privacy preferences are fulfilled. An example of this scenario consists of employees who can be provided with services such as travel offers, ticket booking services, etc. by their company. To benefit from these services, employees need to fill registration forms where they disclose personal data, e.g., name, surname, age, address, etc. These forms also allow employees to specify the conditions under which their personal data can be accessed. As services offered by the company may be provided by entities internal to the company (e.g., a career development advice service) as well as entities external to the company (e.g., a travel agency) these conditions can, for instance, restrict the access to certain personal data items to entities within the company.

3 Related Work

A set of key requirements must be fulfilled in order to provide the data subject the means to control how a data collector uses their personal data items – see Section 1. First, (1) the data subject must be given the means to fully specify the actions that can be performed on their personal data items as well as the conditions under which they can be performed. Then, (2) as specified by Karjoth et al. in [4], after personal data items have been sent to the data collector, the data collector must maintain a link between each of the received personal data items and the associated preferences. This, to guarantee that the conditions under which each personal data item may be accessed can always be identified. This aspect has been dealt with by the PRIME project [6]. However, here we further refine that work by considering both consent and revocation aspects, i.e., we implement the lifecycle management of consent. Finally, (3) the previous conditions must be enforced each time that an access to the associated personal data item is requested. Mechanisms must be put into place that, for each data item and each data subject, check whether the values of the preferences allow the access to be granted.

As previously discussed, the first of the previous requirements (cf. (1)) cannot be fulfilled by traditional opt in/out mechanism. A more suitable approach is the one introduced in the EnCoRe project [7] where the data subject consents to the use of their personal data by specifying, for each personal data item or group of personal data items, fine-grained privacy preferences defining how these data items must be used. This approach has the advantage of coping with situations, generally not dealt with in the literature, where the data subject decides to revoke the right they gave to a data collector to use their personal data. By properly updating the preferences stored by the data collector, the data subject can indeed make some of their personal data items be no longer validly accessible. Such preferences can for instance be: a date *authorised_date* until when a data item can be used by an authorised principal, a list *authorised_third_parties* of third parties to which the data item can be sent, a list *authorised_purposes* of purposes for which the access to the data item can be granted, etc.

Different solutions already proposed in the literature might be considered to fulfil the third requirement (cf. (3)). Hippocratic databases [8], for instance, are a specific type of database which rely on a relational data model to allow access to data to be granted based on ten privacy principles. Using Hippocratic databases to solve the considered problem would require to modify their data model and to make it adaptable to each data collector's requirements. Another strong limitation is that Hippocratic databases do not apply to other types of data repositories than relational databases. The solution proposed by Byun and Li in [9] only deals with purpose-based access control. The Enterprise Privacy Authorization Language (EPAL) [10] could also be considered. It is a language that allows the definition of fine-grained access control policies. As it is considered to be a subset of the XACML standard [11], XACML would suit better the resolution of our problem. However, XACML does not allow specifying, in the access control policies' rules, some conditions which depend on the values of some data stored in some repositories. It indeed only allows policies' rules to contain conditions specified on the "*subject*", "*resource*", "*action*" or "*environment*" attributes concerned by an access request. But none of these attributes corresponds to our privacy preferences. Fine-grained preferences, as proposed in our solution, are not dealt with by the XACML standard or by the XACML privacy profile [12] which only allows to make authorisation decisions based on the purpose for which an access is requested. Consequently, no mechanism is provided that permits XACML to make authorisation decisions based on fine-grained preferences. Casassa Mont et al. proposed in [13] a solution that does provide preference-based access control. However, it relies on a proprietary language. Hence, there is the need to ensure that the same can be achieved with open languages, such as XACML and/or their extensions. Kolter et al. proposed in [14] a solution relying on XACML in which clients specify privacy preferences by defining constrains that a PDP, trusted by the service provider, must fulfil in order this PDP to be chosen by the client to evaluate an access request. Therefore, it does not allow access control decisions to be based on privacy preferences specified by data subjects but only allows some policy to be evaluated by some PDP fulfilling access requester's privacy preferences. In [14] only access requesters' privacy is dealt with while in this paper, the goal is to protect both access requesters' and data subjects' privacy. Besides this, the solution proposed in [14] requires PDPs to be not only trusted to properly evaluate some policies but also to properly manage some received privacy-sensitive attributes, which goes beyond the traditional role of the PDP.

A solution to allow XACML to provide preference-based access control could be to import the preferences within access requests transmitted to the PDP. However, as a request can consist in accessing personal data from very large numbers of data subjects, providing preference-based access control in this case would require incorporating very large numbers of preferences within the request. This does not scale. The foregoing highlights that the XACML language needs to be extended to allow conditions on preferences – stored in some repositories – to be specified within the policies' rules.

As conditions within rules need to be expressed based on preferences' values, evaluating the policies' rules requires the PDP to obtain the value of the preferences during the decision making process. However, good practice requires separating the decision making from the data access. And in the considered case, preferences themselves are personal data- stored in some data repositories.

In the remainder of this paper, we propose an extension to the XACML architecture that solves this problem. Our solution uses some of the concepts of the Identity Governance Framework (IGF) [15] to allow access control decisions to be made based on preferences' values. This, without making any XACML component – i.e., neither the PDP nor the PEP – access the preferences and the personal data items. This guarantees that privacy preferences, as any other personal data, are accessed only by authorised principals.

4 Assumptions

In order to define our solution, we make assumptions that apply to the considered organisation and assumptions that are specific to the proposed approach. The former are realistic as they cover techniques that are already in place in most organisations. The latter have only a marginal impact on existing organisations' identity and access management solutions (IAM) as it requires new components to be added to existing IAM systems and which, we believe, can adapt to legacy systems: this because our solution is independent of the data access protocol used by the data repositories where are stored the personal data items (see Sections 5 for the details). Our assumptions are:

1. **The data collector is an organisation.** This, because most of the time, individuals are required to disclose some personal data when they request an organisation to provide them some services.

2. **A trusted third party (TTP) is available at the data controller.** This TTP manages cryptographic keys and issues certificates to principals at the data collector. Such TTP can be internal to the organisation. If we consider the employee scenario of Section 2, the TTP could be the organisation's human resources service as it has the means to verify employees' identities and therefore to vouch for these employees' identities to third parties.

3. **Principals have encryption and signing capabilities.** They are able to sign messages that they generate and encrypt/decrypt those that they send/receive. In many organisations solutions relying on encryption are deployed which provide employees remote access to these organisations' information systems. Therefore most organisations already have the capabilities to support encryption and signing.

4. **The proposed extended XACML architecture is initialised by some trusted administrators.** At the initialisation of the system, the data collector's system administrator specifies the set of preferences that should be taken into account by the system for each personal data item or group of personal data items to be collected. Subsequently, data subjects are free to specify the value that they wish for these preferences. The policy administrator specifies the policies. It also specifies, in a response formatting file signed with his private key, the format in which the AA should return personal data to requesting principals. As most organisations do have information systems managed by specialists, it is highly probable that the management of policies as well as the initialisation of the system will be achieved by such specialists.

5. **A front-end application is available which allows the data subject to send their personal data items and privacy preferences to the data collector's AA.** The front-end application displays some forms containing fields to be filled in by the data subject with suitable personal data items. The forms also contain some privacy preferences fields permitting the data subject to specify how each personal data item or group of personal data items must be used by the data collector.

6. **A mechanism is in place that permits the data subject and the data collector's AA to negotiate an Attribute Authority Policy Markup Language (AAPML) contract specifying how the personal data items and the preferences should be used by the AA.** The AAPML contract can be consumed by a PEP [16]. It is important to note that we use AAPML for the purpose it has been defined for (see [17]). The definition of AAPML policies is out of the scope of this paper, however examples of AAPML policies can be found in [16, 17].

7. **The data subject discloses their personal data items and privacy preferences to an AA situated at the data collector.**

8. **An extension to the XACML language exists which allows policies' rules to contain conditions expressed on the value of preferences stored in some data repositories.** We do not discuss this extension.

9. **Components are trusted to behave as specified.** We assume that they cannot be tampered with.

5 Proposed extended XACML architecture

As discussed in Section 3, there is the need to define a solution relying on an open standard for access control which allows data collectors to enforce consent based on fine-grained preferences and which fulfil the requirements defined in Section 1. XACML is a standard that specifies an access control architecture relying on the PEP/PDP model introduced in [18] and a rule-based access control language allowing fine-grained access control. Therefore, XACML is a promising candidate to solve our problem. However, as in its current form XACML does not allow access control to be made based on fine-grained privacy preferences used to express conditions within the policies, XACML needs to be extended. Different approaches are possible to achieve this. One of these is to modify the PDP to make it support preference-based access control. However, this would massively impact on existing IAM solutions as they also would have to be redeployed. Here, we propose a solution that is designed to allow access control to be made based on fine-grained preferences without having to heavily change existing IAM solutions. Only minor changes in the message flow are required. The proposed solution further transfers the complexity of providing preference-based access control to specific components located next to the data. Therefore, preference-based access control becomes a modular functionality that can easily be added and removed, as needed, from existing IAM solutions without degrading the security of the services provided by these IAM solutions.

The proposed extended XACML architecture is represented in Figure 1. The first part is the existing XACML architecture. Its role is to make authorisation decisions and to return, when applicable, the requested data to authorised requesting principals. Our extension only impacts on the manner in which the core XACML components interact and on the definition of the access control policies' rules. Indeed, for our solution to be scalable, policies' rules should be expressed in a general enough manner to make them apply to all data subjects' personal data items and preferences. To achieve this, we do not hard code the preferences values within the policies' rules. We only specify the logic relationships that these preferences must verify. The second part is the proposed extension. It is the AA component which is trusted by the data subjects to manage their personal data as they specified. This component provides to the XACML components the minimum information they need to make authorisation decisions based on fine-grained privacy preferences and guarantees, at the same time, that personal data are never disclosed to them. When positive authorisation decisions are returned by the PDP to the AA, the AA extracts the personal data concerned by the response, encrypts them using the access requesters' keys and returns the obtained encrypted personal data to the XACML PEP. The XACML PEP then transmits these data to the access requesters. The AA is composed of two subcomponents: the data repositories and the attribute service (AS). The data repositories store the personal data items and the preferences sent by the data subjects to the data collector. It is important to note that data repositories can rely on different data access protocols [19, 20]. The AS is the component that evaluates how a request, from a component – e.g., XACML component, to access some personal data items or preferences must be dealt with. More specifically, the AS determines – based on the AAPML contract established with the data subject concerned by the request – whether the requested data can be returned to the requesting component. If the data can be returned, the AS identifies the protocols used by the repositories where are stored the data and accordingly formats some requests permitting data to be extracted from each of these repositories. After having obtained the data, the AS determines – using an authenticated response formatting file – the format under which these data must be returned to the requesting component. This is done based on the principle that no personal data must be accessible to any XACML component. Specific formatting of the response, such as additional filtering, may also be specified in the AAPML contract established with the data subject. If it is the case, the filtering is also performed by the AS before the response is returned. Such filtering can, for instance, be the removal of the data subject's Social Security Number from the data items to be returned. Controlling the format under which data have to be returned to a component ensures that components only know the minimum information needed to perform their tasks properly. It therefore permits our extended XACML architecture to run access authorisation processes without ever exposing personal data to unauthorised principals. Three subcomponents help the AA to provide the foregoing. The:

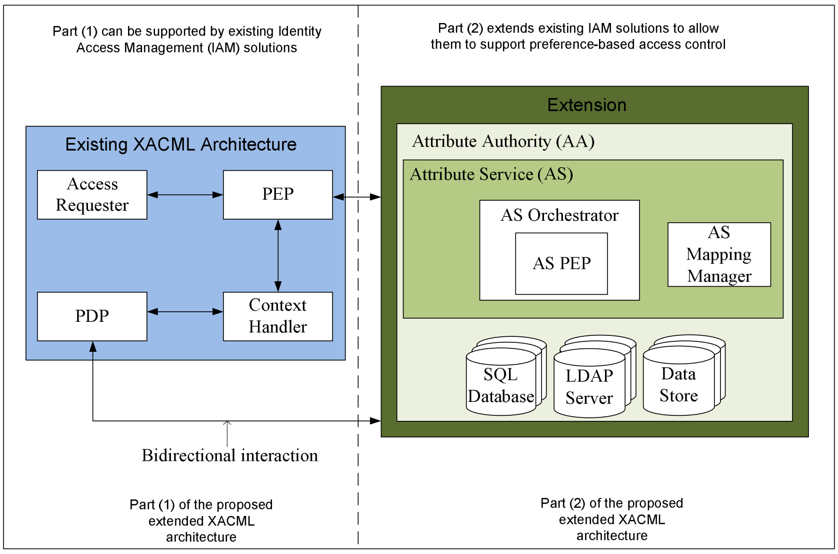
Part (1) can be supported by existing Identity Access Management (IAM) solutions
Part (2) extends existing IAM solutions to allow them to support preference-based access control
Extension
Existing XACML Architecture
Access Requester
PEP
PDP
Context Handler
Attribute Authority (AA)
Attribute Service (AS)
AS Orchestrator
AS PEP
AS Mapping Manager
SQL Database
LDAP Server
Data Store
Bidirectional interaction
Part (1) of the proposed extended XACML architecture
Part (2) of the proposed extended XACML architecture

Fig. 1. Extended XACML architecture and the interactions between its components

- **Authentication Service Policy Enforcement Point.** The AS PEP is a subcomponent of the AS Orchestrator. It stores the AAPML contracts established with the data subjects, based on which it determines whether some personal data items and privacy preferences, stored in the AA's data repositories, can be returned to the AS Orchestrator. Therefore, the AS PEP guarantees that the AA always manages the personal data that it stores as specified by the data subject.

- **Authentication Service Mapping Manager.** The AS Mapping Manager manages the different data representations that are used in the data repositories. It allows the AS PEP to properly format the data access requests to be sent to the data repositories. Therefore, it makes it possible for the proposed extended XACML architecture to be used with data repositories relying on different protocols and to adapt to any legacy system.

- **Authentication Service Orchestrator.** The AS Orchestrator orchestrates the mechanisms that make it possible to deal with access requests made to XACML components or authorisation responses received from XACML components. Its behaviour is constrained by its internal AS PEP that must first authorise an action to be performed on personal data, then access these personal data and return them to the AS orchestrator, for the AS orchestrator to perform the authorised action on the obtained personal data.

The AS Orchestrator relies on an authenticated response formatting file that defines the format of the responses that the AS Orchestrator must return, depending on the nature of the received request and on the requesting principal(s).

The AS Orchestrator receives two types of messages from the XACML components. It receives some property requests from the PDP to verify whether some preferences associated with some personal data items verify the conditions specified in the policies' access control rules. The response formatting file specifies that the AS orchestrator must respond to a property request by: *true* if the conditions are verified and *false* if the conditions are not verified. It also receives positive authorisation responses from the PDP. After receiving a positive response, the AS Orchestrator requests its AS PEP to send it the personal data whose access has just been authorised by the PDP. Once these data have been received, they are formatted as required by the AS Orchestrator before being sent to the XACML PEP.

6 Data flow

The mechanism that allows the PDP to evaluate policies based on preferences is represented in Figure 2 and works as follows. An entity that wants to access some personal data sends an access request to the XACML PEP (cf. (1) in Figure 2).

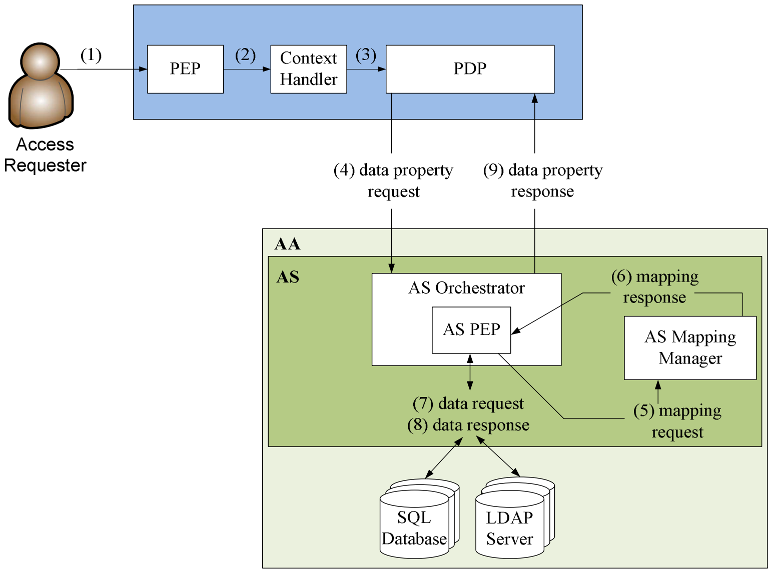

Fig.2. Evaluation of access requests

The process is the same as defined in XACML (cf. (1) to (3) in Figure 2) until the PDP receives the request and evaluates the corresponding policy. As policies contain conditions depending on the value of preferences specified by the data subjects whose personal data need to be accessed (cf. Figure 3), the PDP needs to know whether these conditions are verified for the request being evaluated. For that, the PDP sends a data property request to the AS Orchestrator (cf. (4) in Figure 2) which then requests its AS PEP to return it the suitable preferences. The AS PEP verifies that the AAPML contract, established with the data subject whose preferences need to be accessed, authorises the access. If yes, the AS PEP sends a mapping request to the Mapping Manager (cf. (5)). After having received the AS Mapping Manager's response (cf. (6)), the AS PEP can send some properly formatted data requests to the suitable data repositories (cf. (7)). Once it has received the requested data, the AS PEP returns them to the AS Orchestrator. The AS orchestrator then identifies the format of the response it must return to the PDP and verifies whether the properties requested by the PDP are verified by the data items received from the AS PEP. If it is the case, the AS Orchestrator returns *true* to the PDP and *false* otherwise (cf. (9)). The PDP can then evaluate the policies.

```
<Policy>
    <Rule RuleID="ID" Effect="Permit">
        <Target>
            <Subjects>
                <Subject> authorised role </Subject>
            </Subjects>
            <Resources>
                <Resource> data item to be accessed </Resource>
            </Resource>
            <Actions>
                <Action> read </Action>
            </Actions>
            <Environments>
                <Environment> environment element </Environment>
            </Environments>
            <PreferencesConstraints>
              If
                access purpose == purpose consented by the data subject the data item to be accessed relate to
              and if
                authorised role in authorised requesters specified by the data subject the data item to be accessed relate to
              end if
            <PreferencesConstraints>
        </Target>
    </Rule>
</Policy>
```

This depends on the preferences stored by the AA and specified by the data subject the data item to be accessed relate to

Fig.3. A non-formal example of a policy

After the PDP has rendered its authorisation decision, it sends it to the context handler, as defined in XACML, that then sends it to the XACML PEP (cf. (1) and (2) in Figure 4). Two types of authorisation decisions can be returned to the context handler: "*Deny*" if the access to the requested data has been denied, "*Permit*" if the access to the requested data have been authorised. In the later case, the message sent by the PDP to the context handler may further contain some obligations. When the XACML PEP receives a *Deny* response, it directly sends it to the access requester. However, when the XACML PEP receives a *Permit*, the XACML PEP waits for the AS Orchestrator to send it the data whose access has been permitted.

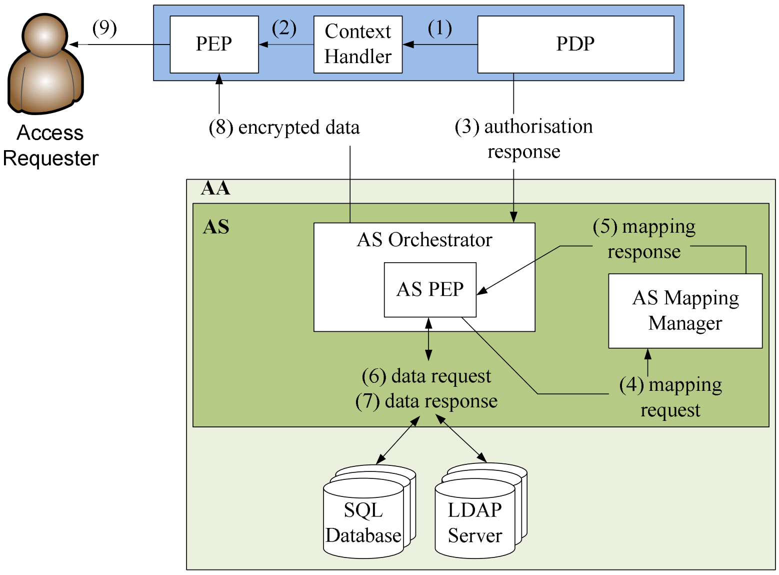

Fig. 4. Management of access authorisation response

After having sent the response to the context handler, the PDP sends the AS Orchestrator an authorisation response (cf. (3)) that contains information about: the data whose access has been authorised, the entity that requested the access and the XACML PEP to which the data item to be extracted must be returned. The AS Orchestrator uses a similar process to the one previously detailed to obtain the data items whose access has been permitted (cf. (4) to (7)). Once the AS Orchestrator does have these data, it encrypts them with the access requester's key and sends them to the XACML PEP (cf. (8)). The PEP then returns the requested data to the access requester (cf. (9)).

7 Current Status

Some of the components, including the XACML PEP and PDP on which our solution relies, have already been implemented. We are leveraging and extending the IGF framework to implement the other aspects of our solution and integrate it with the existing components. The solution proposed in this paper is in the process of being implemented in the context of the EnCoRe project. The final implementation will result in a prototype and a demonstrator which will be used as a basis for testing our approach and future extensions.

8 Conclusion

European data protection regulation mandate that organisations use personal data only as consented by data subjects. In this context, solutions have been proposed to deal with consent management matters, but by only providing generic opt in/out choice. In this paper, we have proposed a solution that extends the XACML architecture so that access control is driven by fine-grained preferences, which represent data subjects' consent. The proposed approach does not require major changes for existing identity access management (IAM) solutions. Only minor changes are required in the message flow. The proposed solution further transfers the complexity of providing preference-based access control to specific components located next to the protected data. Therefore, preference-based access control becomes a modular functionality that can easily be added and removed, as needed, from existing IAM solutions without degrading the security of the services provided by these solutions. The Future work will consist in managing requests to access personal data of many data subjects.

REFERENCES

[1] UK Parliament: Data Protection Act 1998 (1998), http://www.opsi.gov.uk/acts/acts1998/ukpga 19980029 en 1, Accessed the 1 October 2009

[2] The European Parliament and the Council of 24 October 1995: Directive 95/46/EC of the European Parliament and of the Council of 24 October 1995 on the protection of individuals with regard to the processing of personal data and on the free movement of such data (1995), http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=CELEX:31995L0046:EN:HTML, Accessed the 1 October 2009

[3] W3C: The Platform for Privacy Preferences 1.0 (P3P1.0) Specification (2002), http://www.w3.org/TR/P3P/, Accessed on 02 October 2009

[4] Karjoth, G., Schunter, M., Waidner, M.: Platform for enterprise privacy practices: Privacy-enabled management of customer data. In: Privacy Enhancing Technologies. pp. 69–84.Springer (2002)

[5] OASIS: eXtensible Access Control Markup Language (XACML) Version 2.0 (February 2005), http://docs.oasis-open.org/xacml/2.0/accesscontrol-xacml-2.0-core-spec-os.pdf, Accessed on 29 September 2009

[6] Prime project: Prime project website, https://www.prime-project.eu/, Accessed on 26 March 2010

[7] EnCoRe Project: EnCoRe project website, http://www.encore-project.info/, Accessed on 26 October 2009

[8] Agrawal, R., Kiernan, J., Srikant, R., Xu, Y.: Hippocratic Databases. In: Proceedings of the 28th VLDB Conference, Hong Kong, China. pp. 143–154 (2002), http://www.almaden.ibm.com/cs/projects/iis/hdb/Publications/papers/vldb02hippocratic.pdf, Accessed on 02 October 2009

[9] Byun, J.W., Li, N.: Purpose based access control for privacy protection in relational database systems. The VLDB Journal 17(4), 603–619 (2008)

[10] IBM: The Enterprise Privacy Authorization Language (EPAL), EPAL 1.2 specification, http://www.zurich.ibm.com/security/enterprise-privacy/epal/Specification/index.html, Accessed on 02 October 2009

[11] Anderson, A.H.: A comparison of two privacy policy languages: EPAL and XACML. In: SWS '06: Proceedings of the 3rd ACM workshop on Secure web services. pp. 53–60. ACM, New York, NY, USA (2006)

[12] OASIS: Privacy policy profile of XACML v2.0 (February 2005), http://docs.oasis-open.org/xacml/2.0/access control-xacml-2.0-privacyprofile-spec-os.pdf, accessed on 29 September 2009

[13] Casassa Mont, M., Thyne, R., Bramhall, P.: "Privacy Enforcement with HP Select Access for Regulatory Compliance," 2005, accessed on 02 October 2009. [Online]. Available: http://www.hpl.hp.com/techreports/2005/HPL-2005-10.html

[14] Kolter, J., Schillinger, R., Pernul, G.: A privacy-enhanced attribute-based access control system. In: DBSec. pp. 129–143 (2007)

[15]Liberty Alliance Project.: Identity Governance web page, http://www.projectliberty.org/strategic_initiatives/identity_governance, Accessed one 29 September 2009

The original publication is available at http://www.springerlink.com/content/978-3-642-15151-4#section=761233&page=1&locus=0

[16] Hunt, P., Levinson, R.: AAPML: Attribute Authority Policy Markup Language (November 2006), http://www.oracle.com/technology/tech/standards/idm/igf/pdf/IGF-AAPML-spec-08.pdf, Accessed on 30 September 2009

[17] Pohlman, M.B.: Oracle Identity Management Governance, Risk, and Compliance Architecture, Third Edition. Auerbach Publications (2008)

[18] Yavatkar, R. Pendarakis, D., Guerin, R.:“A Framework for Policy-based Admission Control,” RFC 2753 (Informational), Internet Engineering Task Force, January 2000, accessed the 29 September 2009. [Online]. Available: http://tools.ietf.org/pdf/rfc2753.pdf

[19] Zeilenga, K.: Lightweight Directory Access Protocol version 3 (LDAPv3): All Operational Attributes. RFC 3673, http://www.ietf.org/rfc/rfc3673.txt, Accessed on 01 February 2010

[20] Chamberlin, D.D., Boyce, R.F.: Sequel: A structured English query language. In: FIDET ’74: Proceedings of the 1974 ACM SIGFIDET (now SIGMOD) workshop on Data description, access and control. pp. 249–264. ACM, New York, NY, USA (1974)