10.14236/ewic/HIS2015.3

Sticky-Policy enabled authenticated OOXML for Health Care

Grzegorz Spyra
Centre for Distributed Computing Networks and Security
Edinburgh Napier University, Edinburgh, UK
g.spyra@napier.ac.uk

Prof William J Buchanan
Centre for Distributed Computing Networks and Security
Edinburgh Napier University, Edinburgh, UK
w.buchanan@napier.ac.uk

Dr Elias Ekonomou
Centre for Distributed Computing Networks and Security
Edinburgh Napier University, Edinburgh, UK
e.ekonomou@napier.ac.uk

This paper presents a secure medical document sharing model, which addresses confidentiality and authenticity concerns related to cloud-based data protection issues. The paper extends the popular Office Open XML (OOXML) document format with eXtensible Access Control Mark-up Language (XACML) data piece, which defines a sticky-policy and is carried by the document package to enforce data owner access preferences in untrusted networks. Furthermore, it uses Identity Based Encryption (IBE) and Authenticated IBE – two 'next generation' public key cryptographic techniques – to guarantee shared data security. The defined model amends the original IBE construction properties and uses an XACML policy to construct a public key. Using such configuration, the authenticated encryption – with associated data applied to the model – ensures the protection of sensitive data. Shared data is thus encrypted and signed, while the public key (i.e. sticky-policy) is attached to encrypted data and remains in plain text. While the technologies used for the proposed model are not in-themselves new, our novel research contribution lays in combining these technologies in our proposed model.

Sticky Policies, eHealth, IBE, DLP, IRM

1. INTRODUCTION

Several eHealth projects aim to deliver an eHealth platform that would allow health-care institutions to securely host patients' data in a public Internet space. A successful implementation of such a system would facilitate collaboration between medical personnel across different jurisdictions and geo-locations that use various health-care systems. In parallel, economical factors are promoting moving data towards cloud-based solutions, where personal data is not only exposed into the public through Internet channels, but also entirely hosted on semi-trusted cloud-based platforms. Cryptographic techniques used by cloud providers deliver protection from an outsider but not a service provider. Furthermore, recent research shows that even advanced database encryption can be cracked. Not long ago, vulnerabilities allowed a group of researchers to compromise an entire National In-patients Sample (NIS) system security and therefore reveal sensitive personal data [1]. Several works aim to deliver secure solutions ready for the cloud, supporting legacy systems from health-care and educational institutions, enterprise systems and other domains [2-3].

At the same time, the number of exposed vulnerabilities grows with in parallel with newly developed technologies. The complexity of medical systems and health-care processes makes the implementation of single homogeneous e-Health solutions unlikely. Many General Practitioners and private clinics still use Microsoft Office to track patients record, write medical reports and carry out basic administrative tasks. While more dedicated systems can deliver basic data confidentiality and integrity, they may also lack proper data governance, Information Rights Management (IRM) systems or Data Loss Prevention (DLP) programmes. There are various mature products on the market ready to protect sensitive documents hosted in the cloud, including Oracle Information Rights Management (IRM) and Microsoft Retail Management System (RMS) [4]. These solutions, however, have vulnerabilities that intensify when data leave its designated private cloud boundaries.

Trusted Computing is another option, where systems use trusted providers rather than homogeneous solutions. However with fast moving markets and already on-going cloud projects, changing direction toward Trusted Computing would slow down transition into digital eHealth.

Proceedings of BCS Health Informatics Scotland 2015 Conference.
Position Paper.

The solution presented in this paper aims to address sensitive document protection. It leverages Office Open XML (OOXML), an open standard that Microsoft adapted for the MS Office suite [5]. The originality of this approach lays in the fact that it enhances document security by internal structure amendment without making any fundamental changes to an actual document file. The fully protected document benefits from modern public key cryptography. Identity-based encryption (IBE) encrypts the data part, which can then be decrypted and accessed only by an authorized subject. Sticky policies-enabled data can be moved across several cloud platforms and infrastructure boundaries without losing enclosed data integrity and information confidentiality.

This paper is structured as follows: in section 2, we present the sticky-policies paradigm. In section 3, we describe the XACML language used to represent policies and roles. Section 4 discusses OOXML as a document data format and section 5 explains IBE and highlights characteristics that are crucial to complete and secure sticky-policy enabled document. Finally in section 6, we introduce further safeguards to ensure high authenticity for encrypted sensitive personal data.

2. STICKY POLICY

Sticky policies consist of rules defining *who* can access the data, *when, where and how*. Technical roles enforcing Data Protection Act (DPA) [1] directives can be easily combined into policies. Unlike standard policy access control models (see Figure 2), here, the policy sticks to the data and moves along with it, across different untrusted or semi-trusted security boundaries. For example, a sticky policy added to a patient medical report by a data owner (see Figure 3) would cover data owner consent and define any rights to process that data. This way, an access control model implemented via a sticky policy can efficiently secure sensitive Personal Identifiable Information (PII). It can also provide high accountability where single personal data access attempts are subject to extensive auditing [6].

Furthermore, the comprehensive implementation of a sticky policy model also supports several advanced security-auditing functions. Security breaches or data leakage incidents can be identified, collected, and reported for further investigation. Ownership of data released into the Cloud is also strengthened, as the data owner approval policies is associated with the data at all time.

Furthermore, the policy model includes a Trust Authority (TA) predefined by the data owner that specifies where the sticky policy can be interpreted [7]. Information about the TA is attached to the policy and is passed either to local Policy Enforcement Point (on-premises) or to the Service Provider (SP) (i.e. cloud-based e-Health system).

3. XACML

The eXtensible Access Control Mark-up Language (XACML), used in our implementation to represent a sticky policy, easily integrates with Office Open XML (OOXML) document. XACML is a policy standard from OASIS[2]. The policy model defines tuple relationships, where a 'subject' performs a specific 'action' in relation to an 'object'. The policy data model comprises three elements: rule, policy and policy set [8]. A rule (see Figure 1) is the most fundamental policy block. It defines *the target*, ─ which is an object in the access attempt tuple ─, *the effect* ─ which can be expected after evaluation of the rule ─ and *the condition,* representing a Boolean expression (true or false). A Policy Enforcement Point (PEP) enforces stateful conditions and can also include optional advice.

Figure 1: *XACML rule example*

```
<Policy>
  <Rule Effect="Permit">
    <Target>
      <Subject "GROUP(GeneralPractitioner):{956EFF...}"/>
      <Resource "THIS:{8781F074-FAB1-4D5D-BBF0...}"/>
      <Action "Read"/>
    </Target>
  </Rule>
</Policy>
```

Obligation and advice were later distinguished in XACML Version 3 to separate the obligation ─ that is an absolute requirement ─ from non-obligatory advice, which can be considered during access control decision. As an example, we refer to 'advice' if someone could be denied access because he does not have a valid educational '*ac.uk'* email address. The next XACML element, the policy, or rules wrapper, can be passed amongst data-flow entities. It contains the rule-combining algorithm, which defines how composite rules results are combined together.

Finally, the policy set is constructed with the target, the set of policies and the obligation expressions. These are evaluated by the Policy Decision Point (PDP) and passed to the PEP for final policy enforcement actions.

[1] http://www.legislation.gov.uk/UKPGA/1998/29/contents

[2] https://www.oasis-open.org/committees/xacml/

Figure 2: *Standard policies granularly control access over patient's medical record*

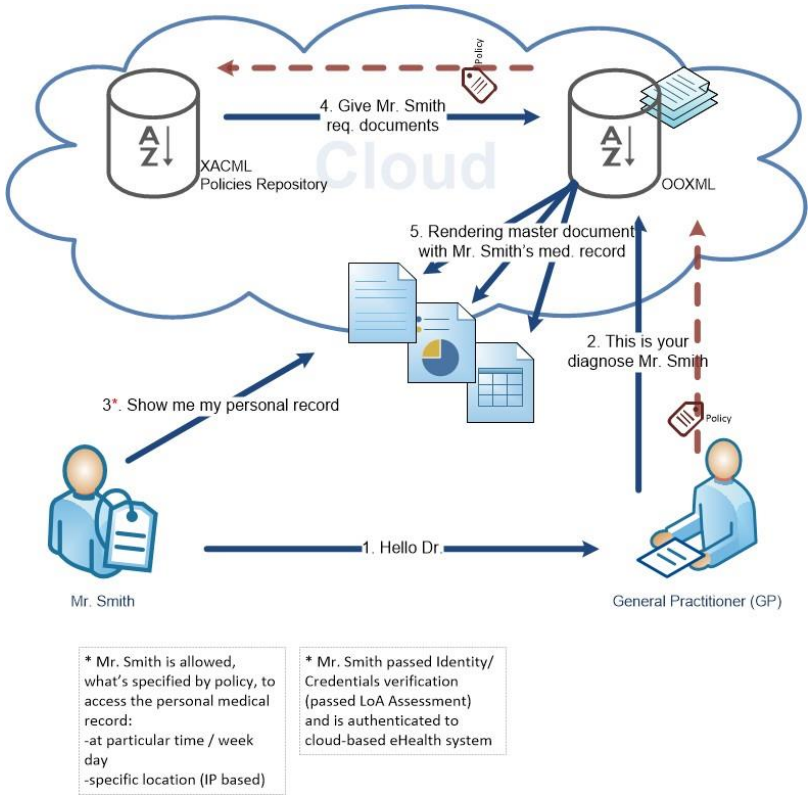
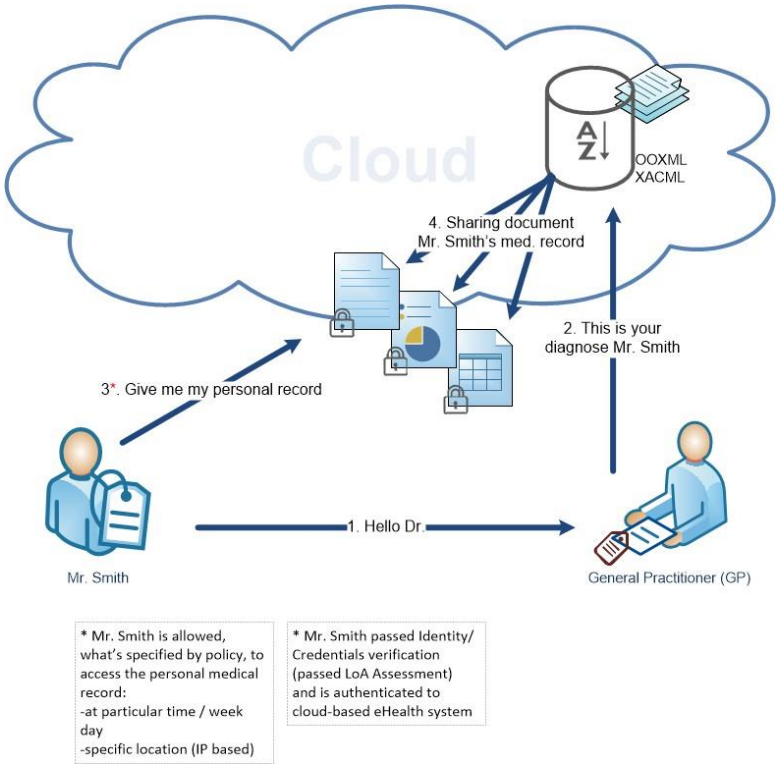


Figure 3: *Sticky-policies granularly control access over patient's medical record*



4. OOXML POLICY WRAPPER

The Office Open XML (OOXML) standard is a popular document standard and is mostly built based on XML files, which refer to each other to form a single document. This format was adapted by Microsoft and used for Microsoft Office. Every modern MS document, spreadsheet, presentation are constructed with OOXML. XML files can be supplemented with other 'reach' files to deliver graphic, multimedia and other elements [5].

OOXML data format can deliver data integrity using internal elements hashing, while confidentiality can be assured by password protection and content encryption through AES 128 or weak RC4 [9]. These techniques are sufficient to protect data that do not leave corporate network, however ─ when leaked ─ this built-in protection may not be sufficient for sensitive personal data. Cloud-based identity meta-data ─ personal information behind digital identity ─ would require additional safeguards from service providers. A securely hosted XML-based information piece would better suit cloud implementations than other data sharing services where data protection is applied on a 'per database' basis [1].

5. IDENTITY-BASED CRYPTOGRAPHY

Shamir in his work on public key encryption scheme suggested that it is possible to build a secure construction where a sender can use a simple text as a public key [10]. In other words, there would be no need to generate a random public and private key pairs because a private key would not have to be generated at the same time when the public key is available and ready to use.

In this crypto model, a sender trying to send a message to a receiver does not have to exchange crypto public keys. The sender can simply take any text known to the receiver and use it as a secure public key. Behind the scenes, both sender and receiver use master key initially generated by a Private Key Generator (PKG). Again, a private key does not have to be generated every time the sender and receiver start a secure conversation.

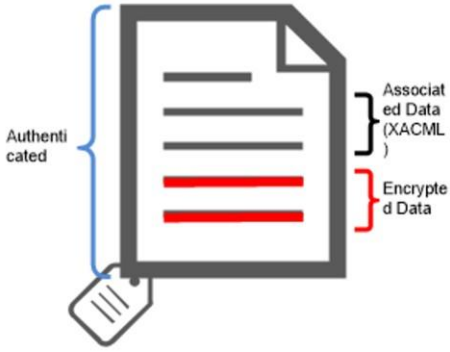
IBE is a special case of Pairing-Based Cryptography (PBC) meeting our prototype requirements. Application of IBE in our model is possible because the XACML policy can be considered as an identity (e.g. email address) and used as a plain text string acting as an identity for the document.

If adverse attempts are made to change the XACML policy once the document has been encrypted and signed, then the Trust Authority will not generate a valid key in the IBE model and thus the document will remain safe.

6. AUTHENTICATED ENCRYPTION WITH ASSOCIATED DATA

Authenticated Encryption with Associated Data (AEAD) was initially introduced as an extension of Authenticated Encryption (AE), where a cryptographic construction could deliver not only message confidentiality but also data integrity. The product of AE authenticated message is only a cypher text, while in AEAD, an encrypted message is accompanied by plain text data that can be used to efficiently evaluate the message authenticity before any other crypto techniques are involved (see Figure 4).

Figure 4: *Secured OOXML document with Sticky-policy attached*

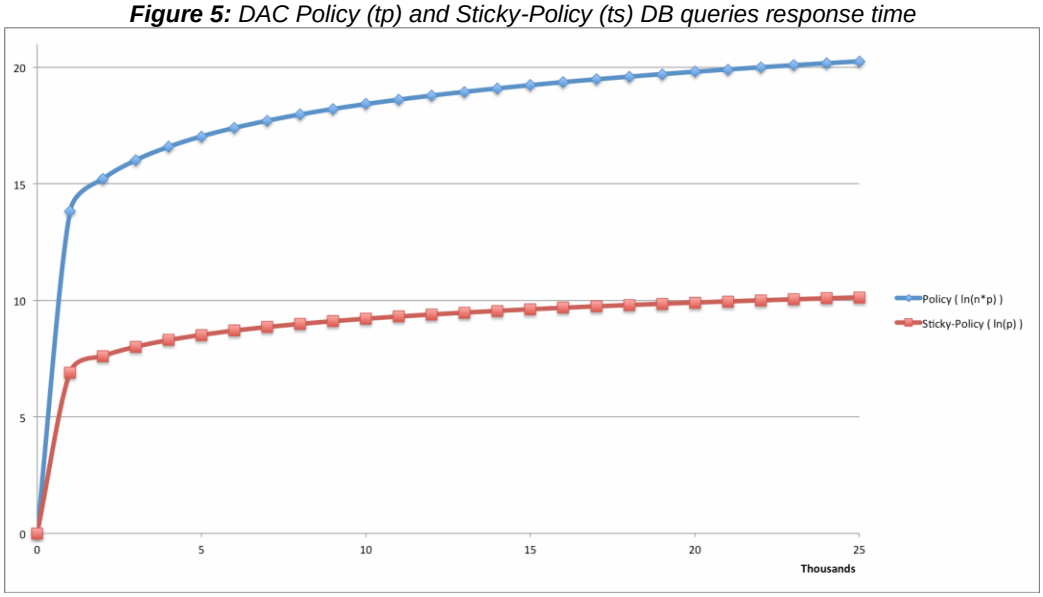


Authenticated Identity-Based Encryption (Authenticated IBE) delivers both message confidentiality and integrity on top of IBE [11]. To empower the security of our model, data integrity should have additional safeguards. This is where we decided to use Authenticated IBE. Our approach uses Authenticated IBE to ensure that both XACML policy and OOXML document content cannot be tampered with. In the Cloud space, only digitally signed documents give a non-repudiation assurance. In other words, the author of a document ─ e.g. a General Practitioner ─ can be sure that ─ once signed ─ the content of a document has not been falsified by a third party [12]. The only bottleneck of Authenticated IBE is that here ─ unlike standard AE ─, the encryption and signing are separate operations. These are therefore more expensive operations although this crypto approach satisfies information security requirements.

7. EVALUATION

Sticky-policies can utilize existing policy frameworks, however an advantage of comprising both a policy and an object (resource) into sticky policy model over keeping the policy separate from the object is the reduced number of model entities and increased DB access performance (see Fig. 5).

Figure 5: *DAC Policy (tp) and Sticky-Policy (ts) DB queries response time*



Having two policy implementations based on transactional databases, it is easy to derive query time assuming it is equal to natural logarithm of total records number. In the policy-based access control model implementation (see Figure 3), the database not only maintains document information ─ which is claimed by the subject ─ but it also holds access policies. Policy can store document location information. However, in access scenario, a subject claims a resource (i.e. a document) based on resource information before this policy is evaluated.

One can calculate query time assuming we have to query policy and document separately (see **Equation 1**).

Equation 1*. Policy and resource query time*

t_p – overall query time
p – number of policies
n – number of documents
a – access control list

$$\forall\, a = p * n$$

$$t_p = \ln(n) = \ln(n) + \ln(n)$$

Equation 2*. Sticky-policy query time*

t_s – overall query time
p – number of policies
n – number of documents

$$\forall\, p = n$$

$$t_s = \ln(n)$$

In sticky-policy model the policy is attached to the resource and both are retrieved in one single request.

We can calculate query time based on a single table query, assuming the policy is encapsulated with a document and both are stored together (see Equation 2).

8. CONTRIBUTION

XACML was designed for authorization to represent access control properties, similar to Access Control Lists (ACL). Used with legacy file systems, the policy can efficiently protect data against unauthorized access. Like in ACL, ─ where permissions are set directly on a file (resource) ─ we used attached sticky-policies to protect information from the moment it leaves legacy security boundaries and reaches cloud systems. Furthermore, OOXML perfectly suits our purpose as it also easily integrates with XACML policy.

We combined sticky policy directly with Microsoft document using the .Net library set. We ran a policy engine against embedded policies. Our work on policy enforcement using system MiniDriver's to intercept low-level document opening instructions and take further access decisions via a policy engine is ongoing.

Regarding data confidentiality and integrity, we are working on modern public key encryption constructions like Identity-Based Encryption. We are currently using Ben Lynn's PBC libraries from Stanford University[3]. We prototyped a Sticky Policy protected document and encrypted it using IBE BF and AES.

[3] https://crypto.stanford.edu/pbc/thesis.html

Furthermore, we compared performance gain from IBE with legacy RSA encryption. We are currently evaluating IBE BF with configuration parameters that would suffice our author XACML policy-based encryption.

9. SUMMARY & FUTURE WORK

Each part of the model proposed here was prototyped and evaluated separately. For sticky-policy enforcement, we used a single Website that can view or edit simplified OOXML content. To fully evaluate the approach described here, we will need a complete on-premise solution implementation that would reside on an end-user machine and control documents opened on the low driver and memory level. Our work on PBC parameters selection is ongoing, however at the moment simplified IBE from Boneh and Franklin Weil Pairing [4] seems sufficient for prototyping. From the data integrity perspective, we will consider further work to leverage Blockchain technology towards high integrity and non-repudiation of medical records and other sensitive data. We are currently focussing on proof-of-security to ensure that an adversary third-party cannot gain advantage and easily break the security of our proposed model.

The combination of XACML, OOXML and Identity-Based Encryption (IBE) and Authenticated Encryption can deliver functionalities for a Cloud-based access control framework where sensitive personal data can be securely stored in a public cloud space (Fig. 2). Any medical institution, GP or private clinic can use the approach described here to introduce new data protection safeguards before data migration into the Cloud.

The solution can be delivered as a cloud service or as a semi 'on-premise' implementation.

In the cloud-based solution, documents can be hosted and edited in the cloud and never stored locally; however, only simplified document formatting can be used.

Semi 'on-premises' solutions require a dedicated service that can enforce security at the user machine and control cloud-based Trust Authority requests.

10. REFERENCES

[1] http://arstechnica.com/security/2015/09/ms-researchers-claim-to-crack-encrypted-database-with-old-simple-trick/

[2] Jain, A., & Farkas, C. (2013). Ontology-Based Authorization Model for XML Data in Distributed Systems. In Digital Rights Management (pp. 210-236). IGI Global. doi:10.4018/978-1-4666-2136-7.ch012

[3] Le, X. H., Doll, T., Barbosu, M., Luque, A., & Wang, D. (2012). An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow. Journal of Biomedical Informatics, 45(6), 1084-1107. doi:10.1016/j.jbi.2012.06.001

[4] Tan, Y. S., Ko, R. K. L., Jagadpramana, P., et al. Tracking of Data Leaving the Cloud. In Proceedings of Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on, pp. 137-144

[5] http://www.ecma-international.org/publications/standards/Ecma-376.htm

[6] Mont, M. C.; Pearson, S. Bramhall, P. Towards Accountable Management of Identity and Privacy: Sticky Policies and Enforceable Tracing Services Proceedings of the 14th International Workshop on Database and Expert Systems Applications, IEEE Computer Society, 2003, 377-

[7] Pearson, S.; Mont, M. C. & Kounga, G. Lee, C.; Seigneur, J.-M.; Park, J. J. & Wagner, R. R. (Ed.) Secure and Trust Computing, Data Management, and Applications: STA 2011 Proceedings Enhancing Accountability in the Cloud via Sticky Policies Springer Berlin Heidelberg, 2011, 146-155

[8] https://www.oasis-open.org/committees/download.php/2713/Brief_Introduction_to_XACML.html

[9] Wu, H.: The Misuse of RC4 in Microsoft Word and Excel. Cryptology ePrint Archive, Report 2005/007

[10] Shamir, A. Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley & D. Chaum (Eds.), Advances in Cryptology (Vol. 196, pp. 47-53). Springer Berlin Heidelberg. (1985).

[11] Lynn, B. (2002). Authenticated Identity-Based Encryption. IACR Cryptology ePrint Archive. Retrieved from http://eprint.iacr.org/2002/072

[12] Abbas, A., & Khan, S. (2014). A Review on the State-of-the-Art Privacy Preserving Approaches in E-Health Clouds. IEEE Journal of Biomedical and Health Informatics, 2194(c), 1-1.

[4] https://crypto.stanford.edu/~dabo/papers/bfibe.pdf