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

Abstract – This paper proposes a secure medical document sharing construction, 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 which is defined with 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, which are two next generation public key cryptographic techniques to guarantee data security. The defined model amends the original IBE construction properties and uses an XACML policy as a public key. With this the authenticated encryption, with associated data concept 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 remains in plain text format.

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. Furthermore successful implementation of such a system should ease collaboration between medical personnel from various health-care systems, across different jurisdictions and geo-locations. In parallel, economical factors are 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. Several works (Jain & Farkas, 2013; Le, Doll, Barbosu, Luque, & Wang, 2012) aim to deliver solutions ready for the cloud, supporting legacy systems from health-care and educational institutions, enterprises and others.

The complexity of medical systems and internal health-care processes often mean that it is unlikely that they can be replaced with a single homogeneous e-Health solution. Many General Practitioners and private clinics still use Microsoft Office to: track patients record; write medical reports; and run administration. While dedicated systems often deliver basic data confidentiality and integrity, without proper data governance or Data Loss Prevention (DLP) programme in place guarantee. There are various mature products on the market ready to protect sensitive documents hosted in the cloud, including Oracle IRM and Microsoft RMS (Tan et al., 2012). These solutions, however, have weaknesses when data leaves its designated cloud boundaries. Trusted Computing is another option, where systems use trusted providers rather than homogeneous solutions. In fast moving markets and really advanced in terms of progress eHealth projects, changing direction toward Trusted Computing would delay all existing plans for couple of years.

The solution in this paper is designed to address sensitive documents protection problem and is based on Office Open XML (OOXML) (Apple et al., 2012), an open standard that Microsoft adapted for the MS Office suite. What makes this approach unique is the fact that it enhances document security by leveraging its internal structure without making any fundamental amendments to actual document file. The fully protected document has the data part encrypted, which can be accessed only by authorized subject, although it can be moved across several cloud platforms and infrastructure boundaries without losing its security information.

In Chapter 2 we discuss the sticky-policies paradigm and in Chapter 3 we describe XACML language we use to represent policies and roles. Chapter 4 discusses OOXML as a document data format and Chapter 5 explains IBE and highlight characteristics that are crucial to complete and

Proceedings of….

secure sticky-policy enabled document. Finally in Chapter 6 we bring further safeguards to ensure high authenticity for encrypted sensitive personal data.

2. STICKY POLICY

Sticky policies consist of rules defining *who*, *when*, *where* and *how* to access the data. Unlike standard policy access control model (see Figure 1), the policy sticks to the data. Sticky policy added to medical report about a patient by data owner (see Figure 2) would cover data owner consent and define any subject rights to process that data. This access control model will secure Personal Identifiable Information (PII) with high accountability where each personal data access attempt is a subject of extensive auditing (Pearson, Bramhall, & HP Laboratories, 2003). Furthermore the complete implementation of sticky policy model supports several security auditing functions. Any security breach or a data leakage incident is reported by sticky policies framework, and can be tracked and lead to legal consequences as policies can be easy combined into technically enforced Data Protection Act (DPA) principals.

The data owners can feel that they own the data released into the Cloud because not only the policies associated with data following data owner approval, but also for the Trust Authority (TA), which specifies where the sticky policy can be interpreted is pre-selected by the data owner (Pearson, Mont, & Kounga, 2011). Information about the TA is attached to the policy as a part of policy derivation path and is passed either to local Policy Enforcement Point (on-premises) or to the Service Provider (SP) (i.e. cloud-based e-Health system).

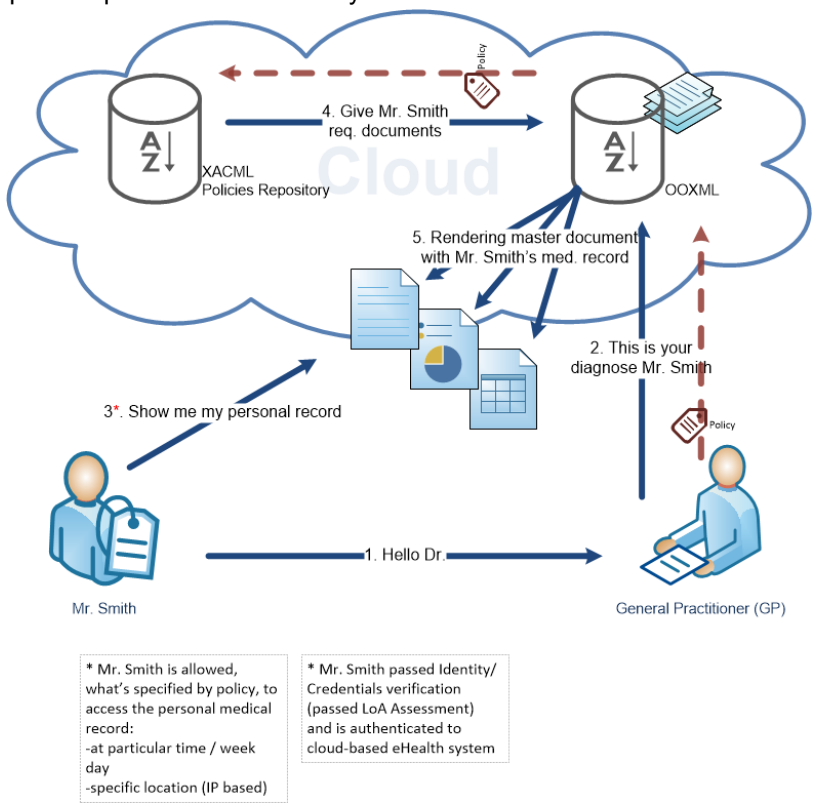

Figure 1: *Standard policies granularly control access over patient's medical record*

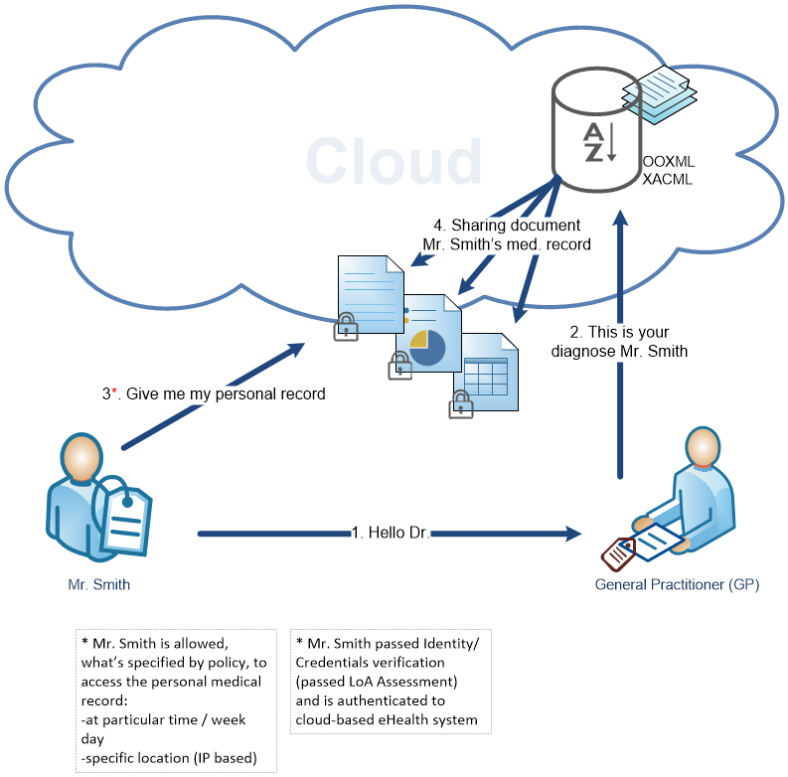

Figure 2: *Sticky-policies granularly control access over patient's medical record*

3. XACML

eXtensible Access Control Mark-up Language (XACML), in our implementation used as a sticky policy language, easily integrates with Office Open XML (OOXML) document. XACML is a policies standard from OASIS, where the policy model defines tuple relationships where subject performs particular action against object. Policy data model comprises of three elements rule, policy and policy set (Saldhana et al., 2013). Rule is the most fundamental piece of policy defining: the target, which is an object in access attempt tuple, the effect that can be expected after evaluation of the rule, the condition representing Boolean expression, the obligation for Policy Enforcement Point (PEP) enforcing stateful conditions and the advice, which is optional (unlike obligations).

The last two obligations and advice were distinguished in XACML version 3 to separate the obligation that is must statement for PEP from the advice that can be considered by PEP, e.g. Bob can be denied access because he does not have valid email address from educational *ac.uk* domain. Policy, the next XACML element, is a rules wrapper that can be passed amongst data-flow entities. It is constructed with a policy target, which will be described later; the rule-combining algorithm defines how the composite rules results are combined together, the obligation, same as in rule it enforces PEP to accept additional conditions and the advice, which has similar role as obligation (see the rule definition). The last XACML element, the policy set is constructed with the target, the set of policies, the obligation expressions, are evaluated by Policy Decision Point (PDP) into obligations and passed onto PEP and the advice expressions which same as obligation expressions are resolved into advices and passed to PEP.

4. OOXML POLICY WRAPPER

Office Open XML (OOXML) standard is mostly built on top of XML files, which reference to each other to form a single document. XML files can be supplemented with other reach files to deliver graphic, multimedia and other elements (Apple et al., 2012). OOXML data format can deliver data integrity using internal elements hashing, while confidentiality can be assured by ZIP wrapper password protection and content encryption. These techniques are sufficient to protect content that does not leave corporate network, however when leaked this built-in protection may not be sufficient for personal data. Cloud-based identity meta-data sharing solution in order to utilize OOXML standard would require additional safeguards from service providers. By hosting and secure delivery of information piece rather than the entire data document, data structured using this XML based standard can be well protected.

5. IDENTITY-BASED CRYPTOGRAPHY

Shamir in (Shamir, 1985) work on public key encryption scheme suggested that it is possible to build a secure construction where sender can use a simple text string as a public key. In other words there is no need to generate random public and private key pairs. Person a sender, trying to send a message to another person - a receiver does not have to exchange crypto public keys but sender can simply take a text known to receiver and use it as a secure public key.

Further works show practical applications of Shamir's concept where Identity-Based Encryption (IBE) from (Boneh & Franklin, 2003) deliver fully satisfactory secure IBE construction. Receiver identity used as a public key should be known to sender, therefore implementations with simple and unique identity format (e.g. email address) complement this crypto structure. Here private key is generated for receiver by Private Key Generator (PKG) delivered by trusted party that can also verify receiver identity.

Important feature of IBE is fact that private key does not have to be generated before data is encrypted. We use this characteristic in our model as unlike in Public Key Cryptography, here is no need to maintain private key vault for every public key. Therefore in case where number of identities (i.e. publicly known text strings) is unlimited, number of key pairs will not have impact on the implementation.

6. AUTHENTICATED ENCRYPTION WITH ASSOCIATED DATA

AEAD was initially introduced as an extension of Authenticated Encryption (AE) where cryptographic construction could deliver not only message confidentiality but also data integrity. Product of authenticated message is a cypher text, while in AEAD an encrypted message data is accompanied by plain text data, although still entire message remains authenticated (see Figure 3).

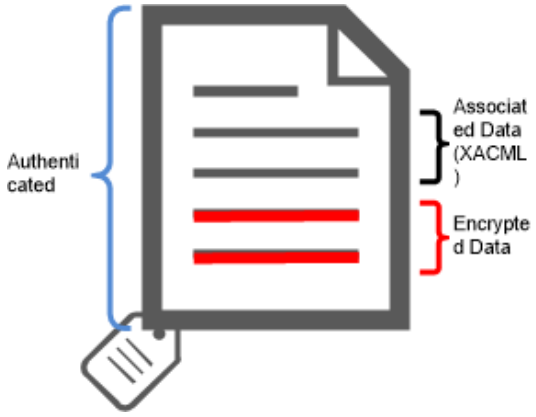

Figure 3: *Secured OOXML document with Sticky-policy attached*

Authenticated Identity-Based Encryption (Authenticated IBE) delivers both message confidentiality and integrity on top of IBE (Lynn, 2002) Unlike standard AE here encryption and signing are separate operations, however this model still satisfies security requirements.

Our approach use Authenticated IBE to ensure that both XACML policy and OOXML document content were not tampered. Only such digitally signed document that leaves local boundaries to be stored in the cloud secures non-repudiation. In other words, author of a document, here General Practitioner can be sure that document content since signed was not amended by any adversary to falsify information.

7. EVALUATION

Sticky-policies can utilize existing policy frameworks, however an advantageous 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 performance (see Figure 4).

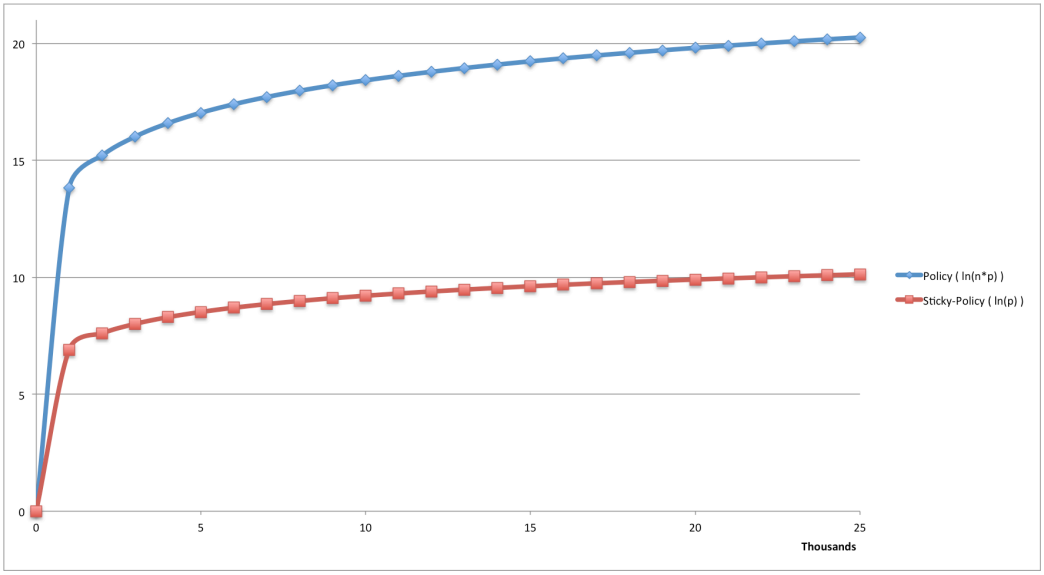

Figure 4: *DAC Policy (tp) and Sticky-Policy (ts) DB queries response time*

Having two policy implementations based on transactional databases it is easy to derive query time assuming it is equal to natural logarithm from total records number. In policy-based access control model (see Figure 2) implementation database maintains not only document information, which is claimed by the subject but also policies. Policy can store document location information, however in access scenario subject claims resource (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).

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

$$\forall\ a = p * n$$
$$t_p = \ln(a) = \ln(n) + \ln(p)$$

Equation 1. Policy and resource query time

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

$$\forall\ p = n$$
$$t_s = \ln(n)$$

Equation 2. Sticky-policy query time

In sticky-policy model the policy is attached to the resource and both are retrieved in one single request. One can calculate query time based on a single table query, assuming policy is encapsulated with a document and are stored together (see Equation 1. Policy and resource query time

).

8. OUR CONTRIBUTION

XACML was designed to represent access control properties for authorization and same as Access Control Lists (ACL) used with legacy file systems it can protect data against unauthorized access. Like in ACL, where permissions are set directly on a file, we used sticky-policy concept to protect information from the moment it leaves legacy security boundaries and reach a cloud. OOXML used by Microsoft perfectly suits our purpose as it easily integrates with XACML due to fact OOXML is a multi XML package file compressed to represent single document file.

Furthermore data confidentiality and integrity can be secured by modern public key encryption constructions like Identity-Based Encryption. IBE is a special case of Pairing-Based Cryptography (PBC) that meets our prototype requirements. XACML policy same as an identity (e.g. email address) passed to IBE is a plain text string. When adversary changed the policy after document was encrypted and signed Trust Authority will not generate a valid secret key therefore document remains safe. To empower security of our model data integrity should have additional safeguards, this is where we decided to use Authenticated IBE.

9. 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 benefit from concept described here authors need complete on-premises solution that would reside on end-user machine and control documents opened on the low driver and memory level. We still work on PBC parameters selection, however IBE seems sufficient for prototype. Authors are currently focused on proof of security to ensure any adversary cannot gain advantage and break security of proposed model.

10. SUMMARY

The combination of XACML, OOXML and Identity-Based Encryption (IBE) and Authenticated Encryption delivers functionality for Cloud-based access control framework where sensitive personal data can be securely stored in a public cloud space (see Figure 2). Any medical institution, GP or private clinic can use approach described here to introduce new data protection safeguards before consider moving into the cloud. Solution can be delivered as a cloud service or semi on-premises. In 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 solution requires dedicated service that can enforce security at the user machine and control cloud-based Trust Authority requests.

11. REFERENCES

Apple, Barclays Capital, BP, The British Library, Essilor, Intel, … the United States Library of Congress. (2012). Information technology — Document description and processing languages — Office Open XML File Formats —Part 1: Fundamentals and Markup Language Reference. Geneva: ISO/IEC. Retrieved from http://standards.iso.org/ittf/PubliclyAvailableStandards/

Boneh, D., & Franklin, M. (2003). Identity-Based Encryption from the Weil Pairing. *SIAM Journal on Computing*, *32*(3), 586–615.

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

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

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

Pearson, S., Bramhall, P., & HP Laboratories. (2003). Towards Accountable Management of Identity and Privacy : Sticky Policies and Enforceable Tracing Services Marco Casassa Mont. In *14th International Workshop on Database and Expert Systems Applications (DEXA'03)*. IEEE Computer Society.

Pearson, S., Mont, M. C., & Kounga, G. (2011). Enhancing Accountability in the Cloud via Sticky Policies. *Secure and Trust Computing, Data Management, and Applications Communications in Computer and Information Science*, *187*, 146–155.

Saldhana, A., Tappetla, A., Anderson, A., Nadalin, A., Parducci, B., Forster, C., … Murdoch, V. (2013). *eXtensible Access Control Markup*.

Shamir, A. (1985). Identity-Based Cryptosystems and Signature Schemes. In G. R. Blakley & D. Chaum (Eds.), *Advances in Cryptology* (Vol. 196, pp. 47–53). Springer Berlin Heidelberg. doi:10.1007/3-540-39568-7_5

Tan, Y. S., Ko, R. K. L., Jagadpramana, P., Suen, C. H., Kirchberg, M., Lim, T. H., … Duc, H. (2012). Tracking of data leaving the cloud. In *Proc. of the 11th IEEE Int. Conference on Trust, Security and Privacy in Computing and Communications, TrustCom-2012 - 11th IEEE Int. Conference on Ubiquitous Computing and Communications, IUCC-2012* (pp. 137–144). Liverpool: IEEE. doi:10.1109/TrustCom.2012.282