Accepted Manuscript

Title: Sticky policies approach within cloud computing

Author: Grzegorz Spyra, William J Buchanan, Elias Ekonomou

PII: S0167-4048(17)30144-X
DOI: http://dx.doi.org/doi: 10.1016/j.cose.2017.07.005
Reference: COSE 1171

To appear in: *Computers & Security*

Received date: 11-5-2017
Revised date: 1-7-2017
Accepted date: 5-7-2017

Please cite this article as: Grzegorz Spyra, William J Buchanan, Elias Ekonomou, Sticky policies approach within cloud computing, *Computers & Security* (2017), http://dx.doi.org/doi: 10.1016/j.cose.2017.07.005.

Sticky policies approach within cloud computing

Grzegorz Spyra, William J Buchanan
The Cyber Academy
Edinburgh Napier University
Edinburgh, UK
{g.spyra, w.buchanan}@napier.ac.uk

Dr Elias Ekonomou
The Cyber Academy
Edinburgh Napier University, Edinburgh, UK
e.ekonomou@napier.ac.uk

Biographical Sketch

Greg Spyra holds a BSc in Network Computing with Distinction from Edinburgh Napier University and has worked for several years in security and technology consulting on worldwide projects. He rejoined the university and completed an MSc in Advanced Security and Digital Forensics in 2012, while working full time as senior security engineer/architect in Switzerland. His Thesis was also rated with distinction in 2012 and so he undertook a PhD position to continue research in Cloud-based Security. He is a member of (ISC)2 security consortium and recently awarded the credential of CISSP. Greg works hard to combine corporate experience with secure technologies of tomorrow. His interest concentrates around Cloud-based Identity and Access Management with intensive focus on eHealth and solutions suitable for currently sensitive markets such as financial, health care and governments.

William J Buchanan is a Professor in the School of Computing at Edinburgh Napier University, and a Fellow of the BCS and the IET. He currently leads the Centre for Distributed Computing, Networks, and Security and The Cyber Academy (http://thecyberacademy.org). His main research focus is around information sharing, such as using Trust and Governance Policies, threat analysis, cryptography, and triage within digital forensics. This has led to several World-wide patents, and in three highly successful spin-out companies: Zonefox (zonefox.com); Symphonic Software (www.symphonicsoft.com/); and Cyan Forensics (cyanforensics.com). Bill regularly appears on TV and radio related to computer security, and has given evidence to the Justice Committee at the Scottish Parliament, along with being part of the BBC Scottish Independence Team of Experts (speciality: Cyber Security). This includes appearances on Newsnight Scotland, Good Morning Scotland, Panorama, and Radio 5 Newsdrive. Bill was named as one of the Top 100 people for Technology in Scotland for 2012, 2013, 2014, 2015 and 2016. In Feb 2016, he was also included in the FutureScot "Top 50 Scottish Tech People Who Are Changing The World". Recently his work on Secret Shares received “Innovation of the Year” at the Scottish Knowledge Exchange Awards, for a research project which involves splitting data into secret shares, and can then be distributed across a public Cloud-based infrastructure. He was included in the JISC Top 50 Higher Education Social Media Influencers in 2015. His current work focuses on the secret share methods, and especially in how documents can be stored securely in public cloud based systems. The current cryptography work around secret shares has won several awards, and is the basis for new funded work. This is further enhanced with work around sticky policies and identity based encryption, and which aims to integrate access rights on document within public cloud systems. This includes the storage of high risk documents, such as health care records and financial information. Along with this he has new research work which integrates machine learning into insider threat detection, and within side channel analysis on embedded systems.

Dr Elias Ekonomou is a Lecturer in the School of Computing at Edinburgh Napier University. Elias joined Edinburgh Napier University in 2011 as a researcher in security for e-Health applications. He worked in the EU FP7 project FI-STAR where he defined the security requirements for the 7 use cases and previously for the clou4health project on the integration of data, identity and access rights across various health domains. Before these experiences, he designed a lightweight yet secure data transportation protocol for Wireless Sensor Networks as part of his PhD research, which he obtained from the University of Salford. He also has 15 years professional experience in website developer and system administration (Unix-like and Windows). He is also a former SE/RSE Enterprise Fellow with experience in productising intellectual property arising from academic research. Elias has also worked in Knowledge Transfer projects with a number of SMEs and

has delivered CPD courses for international corporations, including a top UK bank and public organisations like Police Scotland.

***Abstract*— This paper discusses a secure document sharing approach, which addresses confidentiality, integrity and authenticity concerns related to cloud-based data sharing. This research is focused on a secure construct that would integrate with other cloud ready standards and products for data protection. Sticky policies recently considered as one of the preferred cloud data protection techniques are here combined with standardized OOXML data package. The defined model leverages the Identity Based Encryption (IBE) scheme to attach sticky policies to the data. This paper also shows several security features and functions that are suitable for secure data sharing in the cloud. Technologies used for proposed construction are not new, therefore only their unique combination with AES key derived from XACML sticky policy via IBE and OOXML wrapper constitutes novelty of this research.**

Keywords— IRM, Sticky Policies, IBE, Cloud Security, XACML, Cryptography

1 INTRODUCTION

Cloud service providers (CSP) are offering services that in large organizations and enterprises were previously delivered only on-premises. This introduced completely new challenges for potential CSP customers. Major security organizations offered tough security standards that CSP have to comply with and standards that customers from governmental, financial and public sectors have to implement (Luna et al., 2015). Security standards compliance however is a regulatory form of information security practice not a safeguard that can actually protect the data.

To compete with new technological challenges many data protection services that were previously only delivered within strict security boundaries are offered as a cloud service. One of the major threats is related to data encryption. It has been shown that currently used cryptographic schemes with common setup could be broken with an emerging quantum technology (Chen et al., 2016). Quantum computing has been hanging as sword of Damocles over the cloud security for a decade. No data could be safe but its governance could be improved. Another possible threat is related to Big Data where simple machine learning and business intelligence as a service becomes a way to efficiently process large amounts of anonymized or encrypted personal data. Extracted anonymized data becomes highly sensitive due to fact that illegitimate data analysis applied on a large scale could have potentially a serious social impact (Reimsbach-Kounatze, 2015). Without right dedicated information security techniques for cloud it is rather hard to protect information. Nevertheless, different security techniques having various vulnerabilities could serve better if properly chosen for the right purpose.

CSPs protecting data at rest on per-database basis can leverage database encryption. Recently Microsoft researchers published results around a new efficient homomorphic encryption what could be a solution for a medical data (Dowlin et al., 2015) that has to be processed in a secure manner without divulging underlying information. On the other side just a few months earlier another group of Microsoft researchers proved that database CryptDB encryption, previously acknowledged as a secure data protection technique can be broken with a single trick (Naveed et al., 2015). It might be just a matter of time till homomorphic encryption becomes vulnerable to a new type of attack. Database encryption approach is highly efficient from indexing perspective, however single vulnerability may compromise entire database security and divulge vast amount of sensitive information.

Different, Information Rights Management (IRM) approach applies security on per data piece or data-in-motion basis. CSPs offering IRM as a service encrypt every single document or message separately therefore risk that entire data repository is compromised due to a single vulnerability is reduced. IRM seems to be a relevant and an easy solution for document and message exchange albeit to deliver both security and performance it involves at least two cryptographic methods where keys management becomes complex. There is still a lack of cloud standards and just a few products what makes data exchange between parties homogeneous therefore hard. IRM providers considered new security countermeasures. I.e. Microsoft to improve keys protection, enabled on-premises Hardware Security Module (HSM) support (Sergey Simakov et al., 2015) for its flag, cloud-based IRM product MS Rights Management Services (RMS) Online. IRM itself does not provide functionality for data indexing and its applications without prior data extractions and anonymization are limited in comparison to DB encryption. However still large organizations and enterprises as well as small businesses and individuals store and exchange sensitive information using unprotected documents and single plain text messages. IRM is not suitable for many applications, although is very efficient when it comes to constrained data exchange where two or more parties share a data in asynchronous manner.

In regards to frameworks for cloud data sharing, data hosted and protected by one cloud service provider cannot be securely transferred outside of a single CSP security boundary. Such a migration would require either data to be re-encrypted before migration or cloud providers would have to exchange cryptographic master keys. Cloud data hosting very often is based on storing data by homogeneous application in a public Internet space; what bends initial cloud service principals. Theoretically cloud provider should offer a transparent service that could be dynamically transferred or seized by other cloud service provider without loss of actual service quality and data availability (Leimbach et al., 2014). Using IRM with properly designed infrastructure of distributed trust authorities (TA) could address problem of data sharing and data migration between CSPs and other data sharing parties.

The solution presented here is built on top of existing open standards mainly authentication and authorization framework that have been developed over years and that have been used for various global implementations. Security Assertion Markup Language (SAML) authentication standard developed by The Organization for the Advancement of Structured Information Standards (OASIS) (Cantor et al., 2005) is not discussed further in this paper but it is worth mentioning in regards to actual future implementation. SAML authenticates identity across different organizations and cloud providers. For authorization this research applied an existing eXtensible Access Control Markup Language (XACML) policy framework (Saldhana et al., 2013), which can enforce Discretionary Access Control (DAC) rights and leverage non-DAC roles like in Role-based Access Control (RBAC) systems (Anderson, 2004) or can be combined with modern risk assessment engine (Gasparini, 2013) to control access using dynamic risk calculations. Most of the modern access control techniques can be combined with XACML policy including dynamic membership and rights revocation. Document data format used in this research is based on Office Open XML (OOXML) an open standard (Apple et al., 2006), which defines the XML schemas with conforming vocabularies for word-processing, spreadsheet and presentation documents, as well as the documents packaging. Finally both the XACML-based sticky policy and the data are bound together using Identity Based Encryption (Boneh and Franklin, 2003), a cryptographic primitive that protects the data confidentiality and integrity as well as sticky policy authenticity.

Section 2 shows other works this research refers to and other papers that discuss secure data sharing in the cloud and sticky policies as a cloud enabler. Section 3 describes Identity Based Cryptography and shows its applications for discussed Sticky policies model. Section 4 explains Identity Based Encryption with other entities i.e. sticky policy used for key construction. Section 5 shows IBE security proof that applies to sticky policies model. Section 6 discusses integrity and non-repudiation. Section 7 shows how sticky policies can protect the data and what safeguards it bring.

2 Related Work

Many research projects aim to deliver new data protection safeguards sufficient to protect highly sensitive personal data such as medical or governmental records (Jain and Farkas, 2013; Le et al., 2012), (Abbas and Khan, 2014), (Li et al., 2012).

Sticky policy is not a new data protection model, first discussed in (Karjoth et al., 2002) went through various transformation and has been used for many successful Information Rights Management (IRM) implementations. Existing IRM systems Microsoft Rights Management Services (RMS) Online product (Sergey Simakov et al., 2015) or Oracle Information Rights Management (IRM) (Martin Lambert and Peet, 2010) and other same as sticky policies model attach policy to the data. These are products that have been already evaluated on a larger scale, although none of these products are based on any open standard that could be compatible in the cloud, and actually only RMS was adapted for the cloud use. RMS IRM construct uses eXtensible rights Markup Language (XrML) the rights expression language that was designed for closed environments (Microsoft Corporation, 2008) where could implementation was not considered. XrML (ContentGuard, 2001) and XACML (Saldhana et al., 2013) are very similar, however the semantics used to express access rights are different. Both define tuples where subject is permitted to perform specific activity defined by predicate against access object. In XrML a condition is a functional equivalent of XACML obligations, although what gives XACML advantages are complex expressions and predicates with negative and deny assertions. Note that XrML supports only positive assertion.

XrML defines basic rights and digital licensing where the issuer of the license is effectively an owner of the digital asset. Same as XACML, XrML supports basic cryptographic functions. While XACML leverages eXtensible Stylesheet Language Transformation (XSLT) to constrain the policy, XrML uses templates within a license document. Both standards support external references to policy internal elements. Flexibility of both in terms of elements extensions is similar; these are terms that could be redefined to extend the schema.

Despite of XrML strengths fundamentally it is not suited for complex access policies and rules while XACML is intended to be suitable for a variety of application environments (Saldhana et al., 2013) what perfectly fits various cloud configurations.

XACML itself formulates Attribute-Based Access Control (ABAC) phrases. Its profiles can support different access control models as per (Anderson, 2004), (Gasparini, 2013). This policy-based access control model delivers time constrained access functionality as well as capabilities to handle emergency access requests like in Break-Glass scenario. Cross-Enterprise Security and Privacy Authorization (XSPA) (Mohammad Jafari and DeCouteau, 2014) is a XACML profile dedicated for large enterprise use but mostly for healthcare institutions that exchange information across various security boundaries. Technically the XACML is not a part of any particular cryptographic primitive therefore actual implementation can quickly adapt any new functional requirements. XACML offers various features like JavaScript Object Notation (JSON) (Lockhart and Parducci, 2014) profile to format the policy using light in size attributes representation in compare to heavier XML predecessor. Furthermore it could represent legacy access control objects structure via efficient serialization as described in (Karjoth and Schade, 2013).

Very interesting XACML evaluation has been conducted as a part of Next Generation Access Control (NGAC) model development (Ferraiolo et al., 2016). Other research on secure data sharing in the cloud (Brown and Blough, 2015), (Li et al., 2015), (Pearson et al., 2015) shows that sticky policies approach meets major cloud-based data sharing security objectives. Two works (Brown and Blough, 2015), (Li et al., 2015) discuss similar constructions to sticky policies with IBE and prove its applicability, nonetheless described here use of IBE with XACML sticky-policy as a public key simplifies the proposed method.

Promising approach of using lattice-based cryptography with IBE (Ducas et al., 2014) makes any construct with Identity Based Encryption IBE a quantum computing ready.

Recently widely discussed data protection approach are access models based on Attributes Based Encryption (ABE). Concept itself combines cryptography and elements of access control. Its early implementations (Sahai and Waters, 2005) suffered from dynamic membership control, however later (Boldyreva et al., 2008), (Fan et al., 2014), (Hur and Noh, 2011) ABE was reviewed and empowered with attribute revocation functionality that enabled actually the fundamental access control functions required for cloud-based access management. Each ABE construct (Sahai and Waters, 2005), (Yao et al., 2004), (Fan et al., 2014), (Hur and Noh, 2011), (Yang and Jia, 2014) concentrates on cryptographic operations under several attributes. ABE makes cloud-based authorization a cryptography-centric due to highly constrained implementations by selected ABE primitives. ABE implementations leverage many fundamental access control techniques like Break-Glass (Li et al., 2012) where data could be accessed in an emergency scenario with a post-factum approval or a justification. Also time constrained attributes (Hur and Noh, 2011), (Yang and Jia, 2014) technique that compliments access control system using ABE. Despite of fact that ABE is functionally related to Attribute-Based Access Control (ABAC) model seems it has never been wider discussed in context of standardization to simplify global integration for secure and flexible access control.

3 IDENTITY-BASED CRYPTOGRAPHY

Shamir, in his work on public key encryption scheme (Shamir, 1985), suggested that it is possible to build a secure construction where communicating parties could use a simple text as a public key without a need of keys exchange. Based on this assumption Dan Boneh and Matthew Franklin (Boneh and Franklin, 2003) constructed an efficient fully functional Identity Based Encryption (IBE) scheme using Pairing Based Cryptography (PBC). Proposed solution provided indistinguishability under chosen cipher-text attack, in other words adversary given an encrypted message randomly chosen from a two-element message space could not determine the plain-text message.

Under IBE scheme a message sender can take any arbitrary text known to receiver and use it as a public key. Whereas plain text does not require any further cryptographic safeguards a message receiver requires authentication that proves ownership of the public key, i.e. email address. Using email address as a public key here helps communicating parties to share information about Trust Authority (TA) what constrains the key domain for pairing operations.

Proposed here construction rearranges IBE schema model entities like in (Sahai and Waters, 2005), where Alice does not use Bob's identity only the XACML-formatted sticky policy that identifies the data in a security context is mapped into a public key. Bob does not use his private key but after he is authorized by Trust Authority (TA) policy engine a private key is calculated from the sticky policy and TA master key.

4 IBE-ENABLED STICKY POLICY

Proposed solution combines several Identity-Based Encryption (IBE) and policy framework components into simplified model. Trust Authority (TA) is responsible for policy management as well as for key management. It maintains policy templates (see **Error! Reference source not found.**) for TA responsible contexts, e.g. internal medical templates, private banking customer templates or human resources external candidate templates.

Policy templates together with the TA delivers IBE parameters to the editor application required to generate a policy public key. Based on a policy request the TA also makes an access decision or delegates part of the decision to a third party TA. TA stores master key *{s}* for its domain (see **Error! Reference source not found.**) and after positive access request decision it generates a policy private decryption key that is leased to the policy enforcement editor application. Client or server based editor application handles read or read-write document access based on response from the TA.

How Sticky Policy-enabled OOXML protects files? Authenticated against verified Identity Provider (IdP) Alice in order to protect the document selects preferred Trust Authority (TA) from a list of registered TAs, this way she receives a template of possible policy rules in a given security context (see **Error! Reference source not found.**). After defining policy access rules, the policy set is extended by Alice rights, and together with document global unique identifier and a TA reference the sticky policy is ready to protect the document.

Alice to encrypt the document using IBE BF (Boneh and Franklin, 2003) setup requires policy public key Q_{POL} therefore she generates

$$Q_{POL} = H_1(POL_{ID}), \quad (1)$$

where H_1 is a hash function defined on group $\mathbb{G}_1$ of prime order q such as $H_1: \{0,1\}^* \longrightarrow \mathbb{G}_1^*$, which maps sticky policy POL_{ID} into a single point on an elliptic curve.

Alice generates random r from group $\mathbb{Z}_q = \{0, \dots, q-1\}$ under modulo q and calculates parameters:

$$\begin{cases} U = rP \\ V = e\left(m, H_2\left(\hat{e}\left(R_{pkg}, rQ_{POL}\right)\right)\right) \end{cases}, \quad (2)$$

where e is a symmetric AES encryption function over message m and bilinear map ê. Bilinear map ê over Alice public key Q_{POL} and a TA public key R_{pkg} generated from TA master key. Secret key as per IBE is derived from bilinear mapping ê where $\hat{e}: \mathbb{G}_1 \times \mathbb{G}_1 \longrightarrow \mathbb{G}_2$.

Both AES algorithm and function modulo are symmetric, therefore used in IBE BF (Boneh and Franklin, 2003) modulo operation shown in (3) is replaced with symmetric encryption function (2) e.

$$c = m \oplus H_2\left(\hat{e}\left(R_{pkg}, rQ_{POL}\right)\right), \quad (3)$$

Next, both values U, V are stored inside the OOXML document wrapper and together with embedded sticky policy are shared in the cloud. Parameters same as other OOXML wrapped data should be protected by cloud provider at rest and in transit.

How Sticky Policy-enabled OOXML opens protected file? Authenticated by verified Identity Provider (IdP) Bob accessing the document presents the policy with access request to Trust Authority (TA) using TA reference from sticky policy. TA takes access decision and assuming its positive TA uses secret master key s and computes private key (see **Error! Reference source not found.**) for given sticky policy as follows:

$$S_{POL} = sQ_{POL}, s \in \mathbb{Z}_q \quad (4)$$

Next TA sends policy response together with sticky policy private key S_{POL} to Bob.

Bob can now use symmetric AES decryption function $\mathcal{d}$ on parameter V and hash function $H_2: \mathbb{G}_2 \rightarrow \{0,1\}^n$ and decrypt the document as follows:

$$m = \mathcal{d}\left(V, H_2\left(\hat{e}\left(U, S_{POL}\right)\right)\right) \quad (5)$$

Access right specific decision is made by policy framework based on policy response details, however all possible permissions are interpreted as Read or Read/Write rights.

5 Security of Sticky policies IBE

In evaluated Boneh-Franklin IBE the model security depends mostly on difficulty of solving bilinear Diffie Hellman problem (BDHP) (Boneh and Franklin, 2003) and also correct parameters selection that is must for making discrete logarithm problem (DLP) hard to solve. Based on assumption that probability Pr of finding message m using algorithm $\mathcal{A}$ is negligible it has an advantage ϵ defined as:

$$Pr[\mathcal{A}(P, aP, bP, cP) = e(P,P)^{abc}] \geq \epsilon, \quad (6)$$

Adversary can get an advantage in the selected model if BDHP is easy despite of DLP security.
Having $\{P, aP, bP, cP\} \in \mathbb{G}_1$, find $e(P,P)^{abc}$ Alice encrypts a document m and selects Bob as a receiver using tailored policy under selected trust authority (TA). Document is shared via cloud services and now only Alice – the data owner and Bob should be able to decrypt the data.
Eve, the adversary, illegitimately obtained the protected document and unpacked the U, V parameters, therefore she knows the following:

$$\begin{cases} rP \Leftarrow Q \\ sP = R_{PKG} \\ hP = H_1(POL_{ID}) = Q_{pol} \Leftarrow \exists\, h \in \mathbb{Z}_q, Q_{pol} \in \mathbb{G}_1 \end{cases}, \quad (7)$$

Because of (1), (2), (4), (7) having rP, sP and hP Eve now can derive $\hat{e}(P,P)^{shr}$ from the following:

$$\hat{e}(U, S_{POL}) = \hat{e}\left(rP, sQ_{pol}\right) = \hat{e}(rP, shP) = \hat{e}(P,P)^{rsh}$$

With BDHP easy to solve we compute message m:

$$m = d\big(V, H_2((P,P)^{rsh})\big) \Leftarrow \hat{e}(P,P)^{rsh}$$

what shows that Eve can get an advantage in selected model only if parameters are incorrectly selected.

6 STICKY POLICIES IBE AUTHENTICITY

An adversary cannot tamper a policy attached to the data using proposed method. Acting as a public key the sticky policy is authenticated by IBE scheme. Like in any symmetric or asymmetric encryption only the right key can be used to decrypt the cipher-text. IBE is a public key asymmetric cryptographic primitive therefore for a given public key encrypting the message exists one private key decrypting the cipher-text with this message. If an adversary would try to change the sticky policy attached to the data in this construct after TA authorizes falsified request the received private key cannot be used to decrypt the cipher-text. Adversary having the advantage in a policy engine authorization flow (i.e. Trust Authority) still cannot divulge the message by tampering the sticky policy.

The accepted security notion for the model (Nigel P. Smart et al., 2014) that could provide data non-repudiation assurance with an extra cryptographic operation is an Authenticated Identity-Based Encryption (Authenticated IBE). It delivers both message confidentiality and non-repudiation on top of IBE scheme (Lynn, 2002). To implement this authorship safeguard either sticky policy or OOXML document meta-data should carry information about the data owner. Sender i.e. Alice using own private key can authenticate the encryption. Albeit it requires policy private key being leased by the Trust Authority (TA) during initial encryption.

If data integrity is required there are existing Identity Based Signature (IBS) schemes (Cha and Cheon, 2002). This safeguard is more expensive than non-repudiation as requires separate encryption and signing operations, while Authenticated IBE is even faster than actual IBE encryption. Considering other available technologies for integrity and non-repudiation Blockchain might be a preferred option as unlike signature it verifies data in a historical context (Okupski, 2015). Furthermore, Blockchain service together with Trust Authority (TA) may govern any illegitimate re-encryption attempt of the amended data. Changed document despite of initially defined sticky policy rights giving only Read rights can be rejected by TA therefore change will not be added to the block chain.

Furthermore, a well-defined policy template could highly constrain a sticky policy in a given context as well as enforce requirement to fill an authenticated originator attribute. Depends on the implementation each legitimate amendment made would require re-encryption with a different, new document version and it would require unique identifier of an authenticated change originator. XACML policy defines two safeguards, an obligation and an advice (see **Error! Reference source not found.**). Both could carry further instructions for policy enforcement point behind editor application defining how to handle the initial authorization including basic requirements for data re-encryption under updated policy.

7 STICKY POLICIES AUTHORIZATION

Sticky policies carry authorization information required to protect the data. Unlike conventional policy framework where policy are centrally stored and referenced to data, here policy is attached to the data and follows it to enforce access control rules. Policy evaluation upon access request can check *who you are*, *what you have*, *what you know*, *where you are* and *when and how you can access* the data. E.g. in countries that adapted OECD data protection directives (OECD, 2013) owner consent related to data access can be represented as an access rule and combined into a policy set. As mentioned before data access can be constrained by time. E.g. sticky policy added to a financial report would define any subject rights to process the report within a defined time slot and before or after a specific date.

XACML access request construct represents access tuples, with subject, object and predicate. Subject is the data owner or data processor who wish to access the object. Object is the resource document that can be represented by cloud data hosting provider path and unique data identifier. Predicate defines an action that subject is entitled to based on the policy rules. Because of its internal XML structure XACML policies are defined via attributes represented by name/value pairs. XACML sticky policy subject can be constrained by a technical Role (Anderson, 2004) represented as a group in a target system, where e.g. Role is equal «BusinessEngineering». Because sticky policy remains unencrypted its attribute values could be anonymized or obfuscated as a further safeguard. «BusinessEngineering» Role could be represented by a global unique identifier (see **Error! Reference source not found.**) from within given Trust Authority context. Several Attribute Based Encryption (ABE) works proposed attributes representation using binary-state attributes where attributes unlike in arbitrary-state binary attributes do not directly disclose any information about the content of the protected message. XACML rules may remain in arbitrary-state however in a form that requires attributes mapping to some predefined encoded unique attributes.

XACML policy simply combines Rules, Policies and PolicySets into Policies or PolicySets (see **Error! Reference source not found.**) to protect the resource and enforce access rights defined by data owner. Possibility of Policy and PolicySet nesting gives

many possibilities to represent access conditions, however from architectural perspective it seems reasonable to keep the policy relatively flat and constrained by templates from a given TA context.

XACML policy defines multiple subjects construct with more than one subject involved in access control decision (Saldhana et al., 2013). This technique implements separation of duties security principle. This non-cryptographic safeguard can have functional application similar to Attribute-based Encryption (ABE) or Shamir shared secret (Shamir, 1979) concept. While entire access model document is not cryptographically protected the TA still can reject document access and its decryption if not all policy conditions are met. I.e. document could be accessed only if all subjects agree to open the file.

Interesting functional part that is defined by XACML are obligations and advice. Obligation is a *must* requirement compared to non-obligatory advice, which *can* be considered during access control decision. Obligation is a directive specifying obligatory operation after access request decision. E.g. obligation can instruct to raise a security incident after Eve was denied access to the data. Advice can instruct Bob to use his academic email identity because he does not have a valid educational *ac.uk* domain address. Important feature of both obligation as well as advice is the fact that these can enforce data re-encryption under larger key space or even different cryptographic method.

Data access control implementation based on XACML sticky policy can efficiently secure confidential information and Personal Identifiable Information (PII), provide high accountability where single data access attempt is a subject of auditing (Pearson et al., 2003). Comprehensive implementation of sticky policy model could support advanced security auditing where, security breach or a data leakage incident can be reported and collected, giving significant evidence for further legal investigation. Policy construction is highly simplified with policy templates that could be pre-defined by each TA. Policy template can represent required access evaluation context to be included in the policy.

Policy template will use attribute designators to set the correct rules in the right context.

Constrained XACML policy template simplifies policy generation and reduces complexity on the client side allowing only pre-defied set of rules to be configured.

XACML policy defines which individual or group of individuals in what configuration (i.e. location, time) can be granted permissions to access the protected data in the cloud. Sticky policies implemented based on XACML suffice wide range of access control implementations. This includes modern cloud-enabled authentication and authorization frameworks which leverage SAML and Claims-based authentication like in (Bertocci, 2016) where individual prior to authorization would be a subject of authentication (see **Error! Reference source not found.** and **Error! Reference source not found.**) involving third party Identity Providers (IdP).

8 EVALUATION

Evaluation of proposed secure sticky policy construction was performed on selected functional part of the model in compare to functional equivalent in Microsoft RMS Online service (Sergey Simakov et al., 2015). As discussed earlier sticky policy is used to generate a symmetric key under IBE for AES. In MS RMS the symmetric AES key for data encryption is generated separately. RMS encrypts AES key using RSA as a key wrap, then cipher-text is attached to the data. Therefore, here the evaluation looks only into the initial process of policy setup including AES key protection without actual data encryption (i.e. AES 256). Discussed here MS RMS as well as Oracle IRM both use 2048 as a default RSA asymmetric key size (Sergey Simakov et al., 2015), (Martin Lambert and Peet, 2010) as well as AES with 256 key size. RSA of 3072 and 4096 key sizes are added into evaluation scope as recommended key size for any practical future use as per report (Nigel P. Smart et al., 2014) and also as a required key size if consider more complex large size policies. Considering Information Rights Management (IRM) data access model itself, the cryptographic operations performance of all the crypto primitives from the evaluation would be satisfactory therefore even slower IBE with short pairings requiring more computational time would be acceptable. This part of evaluation is strictly illustrative to correlate IBE as a preferred policy encryption primitive among already used ones key wrap encryption methods.

Evaluation used the Ben Lynn's PBC libraries from Stanford University, while OpenSSL open source project libraries were used for RSA evaluation part. Sticky policy was mapped into a key space using IBE Boneh and Franklin (IBE-BF) scheme. IBE performance was compared to RSA encryption of pseudo randomly generated AES 256 key with no policy attached.

Results were calculated based on 400 different XACML polices of size between 2[KB] and 18 [KB] for IBE and pseudo randomly generated AES 256 symmetric keys for RSA. Results show (see **Error! Reference source not found.**) that both RSA 2048 and RSA 3072 perform better and require less time to complete cryptographic operations of than IBE to pair XACML policy into AES key space.

Evaluation shows that IBE used in our construct performed well compare to RSA even though tests did not calculate policy protection operations. Performance-wise IBE does not stay behind other cryptographic primitives considering fact that it offers similar encryption strength. Furthermore other research claims that PBC performance under specific configuration could be compared to AES 256 (Scott, 2011) therefore it seems that for evaluation the cryptographic operations performance will not be the main concern. The most important IBE advantage over RSA is to what extent it can simplify the IRM model. The process of attaching the access policy and encrypting symmetric key using RMS, which implements RSA in compare to described here IBE-enabled sticky policy is complex, non-generic and product specific.

By looking at 400 arbitrary-state policies (see **Error! Reference source not found.**) defining different rules in medical context for external and internal use, all verified with valid XACML request and response message we could see that any potential symmetric key wrap with asymmetric encryption would require rather large keys. Serializing 400 policies from standard eXtensible Markup Language (XML) format into size-wise lighter JavaScript Object Notation (JSON) shows that rich rule expression that would suffice cloud-based implementation requires larger policy sizes. Policy cannot be protected with asymmetric key, therefore it requires either standard symmetric key encryption or could apply some more complex construct that leverages both symmetric and asymmetric encryption (Sergey Simakov et al., 2015).

9 CONTRIBUTION

Security considerations for sticky policies are important for any sensitive data protection model where parties need to securely exchange the data across different security boundaries. This research compliments various projects aiming to securely bring privacy protection and cloud together, especially health-care strategy of moving into a cloud. It contributes to medical data sharing frameworks recently prototyped and evaluated approaches that leverage sticky policies model. It simplifies data encryption from the proposed frameworks by providing novel application of Identity Based Encryption scheme for sticky policies. Many sticky policy based solutions propose use of key-wrap and policy-wrap model but here policy remains obfuscated in plain text attached to a data. This gives much more flexibility and allows access control decisions to be taken before any cryptographic operations are made.

While other existing rights management systems inherit the best security safeguards from their legacy predecessors, this approach is designed for the cloud use and therefore could adapt selected modern security techniques that were invented to overcome limitations of legacy data protection systems.

10 FUTURE WORK

Each part of the proposed model here was prototyped and evaluated separately. Research work requires model workflows to perform a large-scale evaluation with complete model simulation proving that global implementation will not be an overhead for Trust Authority from IBE parameters and key management perspective. For better evaluation it is recommended to deliver an on-premise solution implementation that would reside on an end-user machine and control documents would be opened on a low driver and memory level. This work would be focused on policy enforcement using system MiniDriver's to intercept low-level document opening instructions and take further access decisions via policy engine.

IBE and RSA performance evaluation could be extended by Microsoft RMS policy encryption, however this requires many policy samples from different templates while this would have no significant impact on measured cryptographic operation times. Pairing Based Cryptography (PBC) parameters selection is still an open topic. Latest IBE schemes like IBE over Lattices (Ducas et al., 2014) could highly improve the model security, and while simplified IBE from Boneh and Franklin Weil Pairing seems sufficient for prototyping, an actual implementation with only asymmetric encryption will require other IBE schemes. From data integrity perspective it might be worth but not necessary to consider a research around blockchain technology towards high integrity and non-repudiation of data in specific contexts.

11 SUMMARY

There are many security gaps that need to be analysed before any cloud data sharing framework can be used to host sensitive data. This paper shows how to address some of the threats and how the presented solution could protect the data. Model simplicity might be very attractive for implementations that look for a simple solution with a potential to be extended by either additional

cryptographic primitive or various obfuscations and encoding methods. Authors suggest that AES is only one of many possibilities for data encryption and under other IBE schemes the well-designed sticky policies model might become even a simpler solution.

Using unencrypted access policy attached to the data helps identify the data as well as the owner in the cloud space. Trivial attributes encoding and anonymization should be sufficient to protect the policy from malicious use. By introducing well-defined policy templates, the policy controls when the data could be accessed, when the data should be re-encrypted with latest TA public key or when the data processor should renew IBE parameters for a given TA.

By keeping authorization and authentication components separate from the actual cryptographic operations on data unlike in ABE, allows a faster development of existing models and easier integration with existing frameworks that are already used for large-scale implementations. Existing ABE-based implementations if used seem over-engineered and therefore still need to reach the maturity level sufficient for large and complex environments.

Furthermore, MS RMS Online is an IRM product that requires existing data governance and for more moderate large organisations that care about data protection is still usually deployed only on-premises. This IRM system has legacy XrML policy format limitations and therefore may not be sufficient for large-scale cloud-based access management systems. MS RMS Online introduced various features like One Time Passwords (OTP) and Hardware Security Module (HSM) what helps companies with secure cloud on-boarding, however there are clear technical limitations that would require further research and development. Nevertheless, MS RMS is not an open standard and there are IRM market competitors. There seems to be no single secure data sharing service that would suit large organizations and enterprises as well as small businesses. Sticky policies with IBE implementation when clearly defined can become a good open alternative for the existing IRM market.

In regards to actual implementation of XACML entities and XACML policy generation Axiomatics and WSO_2 have products and set of stable libraries ready to combine with existing access management systems. Oracle formally Sun Microsystems, Quest Software and other larger software companies have also own XACML implementations, however these are either discontinued or closed for integration with third party systems. To develop a OASIS XACML complaint product is highly simplified due to various open projects with source code available via public code repositories.

To introduce other less functional and more security related techniques, some of the fundamental policy engine and key management components were not discussed here.

REFERENCES

Luna J, Suri N, Iorga M, Karmel A. Leveraging the Potential of Cloud Security Service-Level Agreements through Standards. IEEE Cloud Comput 2015;2:32–40. doi:10.1109/MCC.2015.52.

Chen L, Jordan S, Liu Y-K, Moody D, Peralta R, Perlner R, et al. NISTIR 8105 Draft - Report on Post-Quantum Cryptography. 2016.

Reimsbach-Kounatze C. The Proliferation of " Big Data " and Implications for Official Statistics and Statistical Agencies Christian Reimsbach-Kounatze. 2015. doi:10.1787/5js7t9wqzvg8-en.

Dowlin N, Gilad-Bachrach R, Laine K, Lauter K, Naehrig M, Wernsing J. Manual for Using Homomorphic Encryption for Bioinformatics. Microsoft Research; 2015.

Naveed M, Kamara S, Wright C V. Inference Attacks on Property-Preserving Encrypted Databases. Proc. 22nd ACM SIGSAC Conf. Comput. Commun. Secur. - CCS '15, New York, NY, USA: ACM New York, NY, USA; 2015, p. 644–55. doi:10.1145/2810103.2813651.

Sergey Simakov, Sieber M, Norden M. Azure RMS Security Evaluation Guide. Microsoft; 2015.

Leimbach T, Hallinan D, Bachlechner D, Weber A, Jaglo M, Hennen L, et al. Potential and Impacts of Cloud Computing Services and Social Network Websites. 2014.

Cantor S, Kemp J, Philpott R, Security RSA, Hughes J, Origin A, et al. Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0. 2005.

Saldhana A, Tappetla A, Anderson A, Nadalin A, Parducci B, Forster C, et al. eXtensible Access Control Markup Language (XACML) Version 3.0. 2013.

Anderson A. XACML Profile for Role Based Access Control (RBAC), Version 2.0. vol. 1. 2004.

Gasparini L. XACML and Risk-Aware Access Control. Aberdeen: 2013.

Apple, Barclays Capital, BP, The British Library, Essilor, Intel, et al. Information technology — Document description and processing languages — Office Open XML File Formats — Part 2: Open Packaging Conventions 2006:138.

Boneh D, Franklin M. Identity-Based Encryption from the Weil Pairing. SIAM J Comput 2003;32:586–615.

Jain A, Farkas C. Ontology-Based Authorization Model for XML Data in Distributed Systems. Digit. Rights Manag., IGI Global; 2013, p. 210–36. doi:10.4018/978-1-4666-2136-7.ch012.

Le XH, Doll T, Barbosu M, Luque A, Wang D. An enhancement of the role-based access control model to facilitate information access management in context of team collaboration and workflow. J Biomed Inform 2012;45:1084–107. doi:10.1016/j.jbi.2012.06.001.

Abbas A, Khan S. A Review on the State-of-the-Art Privacy Preserving Approaches in E-Health Clouds. IEEE J Biomed Heal Informatics 2014;2194:1–1. doi:10.1109/JBHI.2014.2300846.

Li M, Yu S, Zheng Y, Ren K, Lou W. Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption. IEEE Trans PARALLEL Distrib Syst 2012;XX:1–14.
Karjoth G, Schunter M, Waidner M. Platform for Enterprise Privacy Practices: Privacy-Enabled Management of Customer Data. In: Dingledine R, Syverson P, editors. Priv. Enhancing Technol., vol. 2482, Springer Berlin Heidelberg; 2002, p. 69–84. doi:10.1007/3-540-36467-6_6.
Martin Lambert, Peet A. Oracle Information Rights Management 11g – Managing information everywhere it is stored and used. Management 2010:23.
Microsoft Corporation. XrML 2008;747717:2–3.
ContentGuard. eXtensible rights Markup Language (XrML) 2 . 0 Specification Part II : Core Schema 2001:1–46.
Mohammad Jafari, DeCouteau D. Cross-Enterprise Security and Privacy Authorization (XSPA) Profile of SAML v2 . 0 for Healthcare Version 2 . 0 Committee Specification Draft 01 /. 2014.
Lockhart H, Parducci B. JSON Profile of XACML 3.0 Version 1.0 2014:1–34.
Karjoth G, Schade A. Serialization of XACML policies. US8458764 B2, 2013.
Ferraiolo D, Chandramouli R, Kuhn R, Hu V. Extensible Access Control Markup Language (XACML) and Next Generation Access Control (NGAC). ABAC '16 Proc. 2016 ACM Int. Work. Attrib. Based Access Control Pages 13-24, New York, NY, USA: ACM New York, NY, USA; 2016, p. 13–24. doi:10.1145/2875491.2875496.
Brown J, Blough DM. Distributed Enforcement of Sticky Policies with Flexible Trust. High Perform. Comput. Commun. (HPCC), 2015 IEEE 7th Int. Symp. Cybersp. Saf. Secur. (CSS), 2015 IEEE 12th Int. Conf. Embed. Softw. Syst. (ICESS), 2015 IEEE 17th Int. Conf., New York: IEEE Xplore; 2015, p. 1202–9. doi:10.1109/HPCC-CSS-ICESS.2015.235.
Li S, Zhang T, Gao J, Park Y. A Sticky Policy Framework for Big Data Security. 2015 IEEE First Int. Conf. Big Data Comput. Serv. Appl., Redwood City: IEEE Xplore; 2015, p. 130–7. doi:10.1109/BigDataService.2015.71.
Pearson S, Reed A, Mont MC, Kounga GLD, Chen L. Policy-based data management US 9203621 B2. US9203621 B2, 2015.
Ducas L, Lyubashevsky V, Prest T. Efficient Identity-Based Encryption over NTRU Lattices. ASIACRYPT 2014 2014;8874:22–41. doi:10.1007/978-3-662-45608-8_2.
Sahai A, Waters B. Fuzzy identity-based encryption. Annu Int Conf Theory … 2005:457–73.
Boldyreva A, Goyal V, Kumar V. Identity-based encryption with efficient revocation. Proc 15th ACM Conf Comput Commun Secur - CCS '08 2008:417. doi:10.1145/1455770.1455823.
Fan CI, Huang VSM, Ruan HM. Arbitrary-state attribute-based encryption with dynamic membership. IEEE Trans Comput 2014;63:1951–61. doi:10.1109/TC.2013.83.
Hur J, Noh DK. Attribute-Based Access Control with Efficient Revocation in Data Outsourcing Systems. IEEE Trans Parallel Distrib Syst 2011;22:1214–21. doi:10.1109/TPDS.2010.203.
Yao D, Fazio N, Dodis Y, Lysyanskaya A. ID-based encryption for complex hierarchies with applications to forward security and broadcast encryption. Proc. 11th ACM Conf. Comput. Commun. Secur. - CCS '04, New York, New York, USA: ACM Press; 2004, p. 354. doi:10.1145/1030083.1030130.
Yang K, Jia X. ABAC: Attribute-Based Access Control. Secur. Cloud Storage Syst., New York, NY: Springer New York; 2014, p. 39–58. doi:10.1007/978-1-4614-7873-7.
Shamir A. Identity-Based Cryptosystems and Signature Schemes. In: Blakley GR, Chaum D, editors. Adv. Cryptol., vol. 196, Springer Berlin Heidelberg; 1985, p. 47–53. doi:10.1007/3-540-39568-7_5.
Nigel P. Smart, Rijmen V, Warinschi B, Watson G. Algorithms, Key Sizes and Parameters Report. Heraklion: 2014.
Lynn B. Authenticated Identity-Based Encryption. 2002.
Cha JC, Cheon JH. An Identity-Based Signature from Gap Diffie-Hellman Groups. Int Assoc Cryptologic Res 2002:18–30. doi:10.1007/3-540-36288-6_2.
Okupski K. Bitcoin Developer Reference. Eindhoven, The Netherlands: 2015.
OECD. Recommendation of the Council concerning Guidelines governing the Protection of Privacy and Transborder Flows of Personal Data (2013) 2013:11–37.
Shamir A. How to share a secret. Commun ACM 1979;22:612–3. doi:10.1145/359168.359176.
Pearson S, Bramhall P, HP Laboratories. Towards Accountable Management of Identity and Privacy : Sticky Policies and Enforceable Tracing Services Marco Casassa Mont. 14th Int. Work. Database Expert Syst. Appl., IEEE Computer Society; 2003.
Bertocci V. Modern authentication with Azure Active Directory for web applications. Redmond, Washington: Microsoft Press; 2016.
Scott M. On the efficient implementation of pairing-based protocols. Cryptogr Coding 2011:296–308. doi:10.1007/978-3-642-25516-8_18.

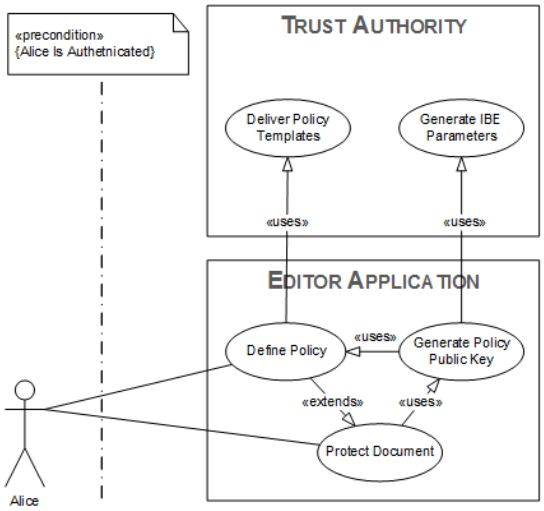

Fig. 1. Sticky Policy IBE Encryption

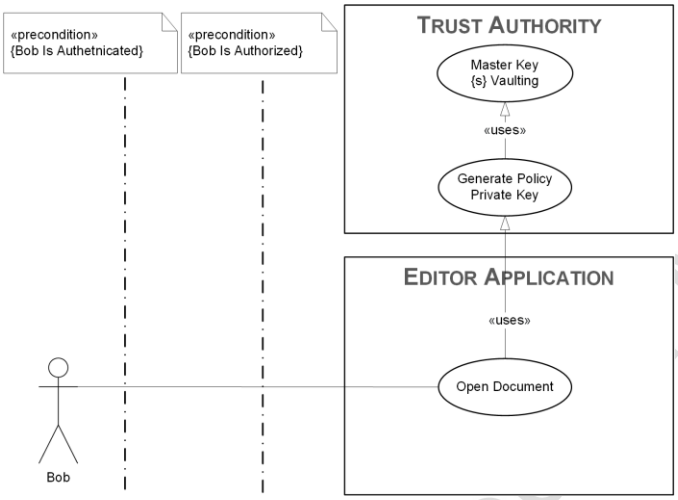

Fig. 2. Sticky Policy IBE Decryption

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

Fig. 3. XACML Rule example

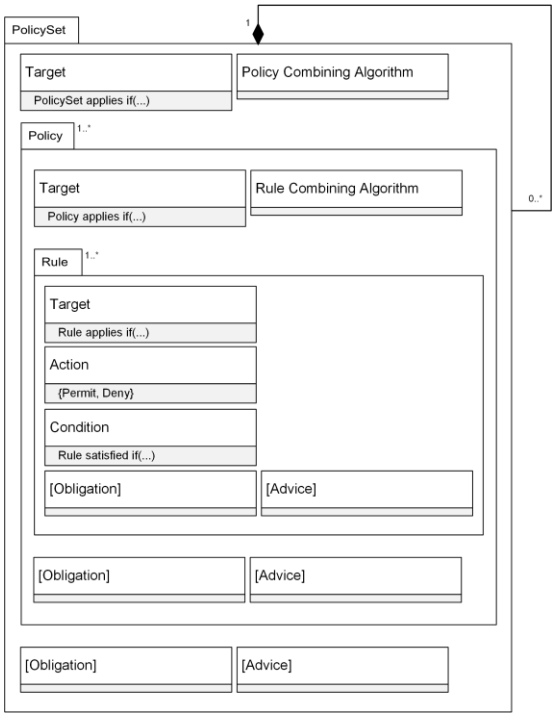

Fig. 4. XACML Policy Construct

```
<Apply xsi:type="AtLeastMemberOf"
functionId="urn:oasis:names:tc:xacml:1.0:function:string-at-least-
one-member-of">
  <Apply
functionId="urn:oasis:names:tc:xacml:1.0:function:string-bag">
            <AttributeValue ParameterId="location"
DataType=http://www.w3.org/2001/XMLSchema#string/>
  </Apply>
  <AttributeDesignator AttributeId="http://schemas.tscp.org/2012-
03/claims/ISO-3166-2"
DataType="http://www.w3.org/2001/XMLSchema#string" />
</Apply>
```

Fig. 5. Policy Template part for location-based access rule

```
<Parameter ParameterId="location">
  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">UK</
AttributeValue>
  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">PL</
AttributeValue>
  <AttributeValue
DataType="http://www.w3.org/2001/XMLSchema#string">CH</
AttributeValue>
</Parameter>
```

Fig. 6. Policy template data representation defining access location in ISO 3166-2 for attribute designator

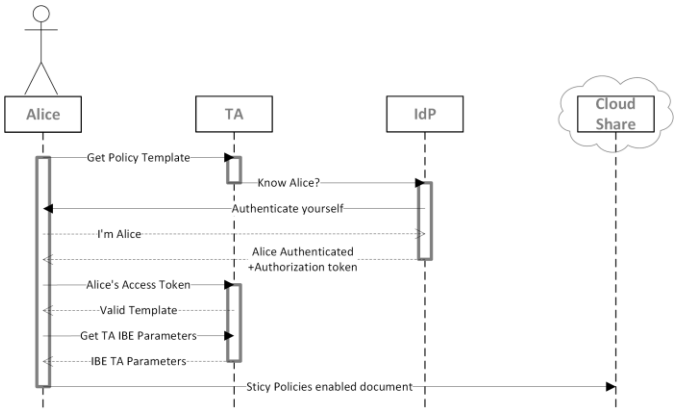

Fig. 7. Sticky Policy IBE Secure Sharing

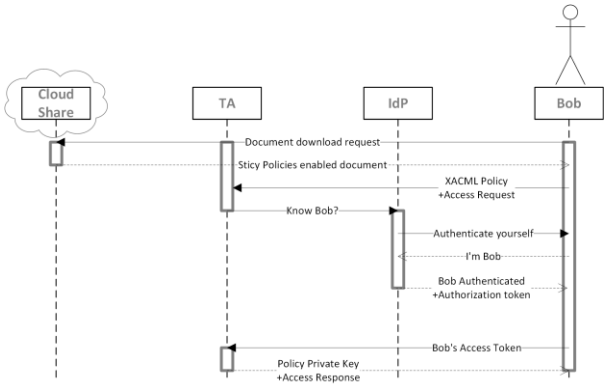

Fig. 8. Sticky Policy IBE Secure Access

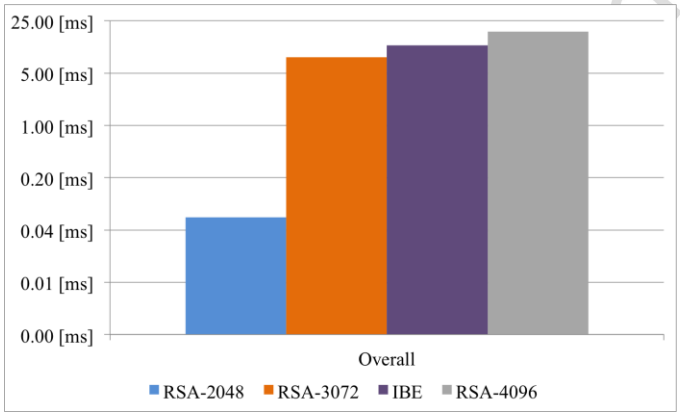

Fig. 9. Times of Sticky policy mapping into AES key space using IBE-BF compared to 2048, 3072 and 4096 RSA operations applied to AES 256 key

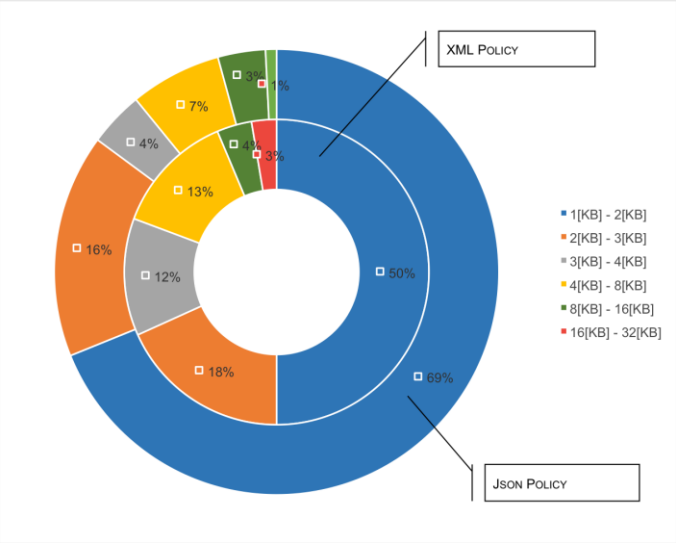

Fig. 10. Policy sizes comparison formatted with XML and Json. XACML policies.