FOI

Object-Based Security with Attribute-Based Encryption

A feasibility study

LARS WESTERDAHL, AMUND GUDMUNDSON HUNSTAD,
FREDRIK MÖRNESTEDT

FOI-R--4002--SE
ISSN 1650-1942 December 2014

Lars Westerdahl, Amund Gudmundson Hunstad,
Fredrik Mörnestedt

Object-Based Security with Attribute-Based Encryption

A feasibility study

Bild/Cover: (Amund Gudmundson Hunstad)

Titel	Objektbaserad säkerhet med attribut-baserad kryptering: En lämplighetsstudie
Title	Object-Based Security with Attribute-Based Encryption: A feasibility Study
Rapportnr/Report no	FOI-R--4002--SE
Månad/Month	December
Utgivningsår/Year	2014
Antal sidor/Pages	52
ISSN	1650-1942
Kund/Customer	Försvarsmakten
Forskningsområde	4. Informationssäkerhet och kommunikation
FoT-område	Ledning och MSI
Projektnr/Project no	E36059
Godkänd av/Approved by	Christian Jönsson
Ansvarig avdelning	Informations- och aerosystem

Sammanfattning

Objektbaserad säkerhet (Obs) är en vision om informationsobjekt som bär med sig ett skydd som bevarar dess sekretess, korrekthet och tillgänglighet. Vanliga klient – serverlösningar kan tillgodose krav på åtkomstkontroll men kräver att konsumenten av informationen står i kontakt med källan för informationsobjektet och dess behörighetskontrollsystem. Enligt Obs bör en behörighetsfunktion fungera utan uppkoppling mot en sådan kontrollfunktion. Ett informationsobjekt som överförs till en konsument via exempelvis ett externt lagringsmedium medför att konsumenten har tillgång till själva informationsobjektet. Konsumenten måste dock enligt Obs fortfarande påvisa sin behörighet innan informationsobjektet kan läsas.

I denna rapport redovisas en studie vilken syftar till att identifiera en teknisk kandidat för att uppfylla Obs. Krypteringsmetoden Attributbaserad kryptering (ABE) analyseras utifrån ett antal frågeställningar som har identifierats utifrån Obs behov. I rapporten föreslås även en konceptuell arkitektur av ett OBS-system baserat på ABE.

Nyckelord: Objektbaserad säkerhet, Attributbaserad kryptering, Åtkomstkontroll

Summary

Object-Based Security (OBS) is a vision of information objects being able to carry with them a protective capability that preserves the properties confidentiality, integrity, and availability. Regular client – server solutions can fulfil the requirements for access control but requires that the consumer of the information is connected to the source of the information object. According to OBS, an access control function should provide service even if there is no connection to such a function. An information object which is transferred to a consumer, for instance via a thumb drive, provides the consumer with access to the actual information object but would, according to OBS, still need to be authorized before accessing the content.

In this report, a study of a technical candidate for the fulfilment of OBS is presented. The encryption method Attribute-Based Encryption (ABE) has been analysed through a set of questions which identify the OBS needs. In the report, a conceptual architecture is proposed which shows how ABE can be utilized to achieve an OBS solution.

Keywords: Object-Based Security, Attribute-Based Encryption, Access Control

Index

1 Introduction

Information technology (IT) has become a natural part of our day-to-day life, both professionally and privately. The maturity of the technology has also affected how it is being used. Information is not just sought and retrieved from the producer of the information; it is just as likely that information is retrieved from an intermediary or a third party. The manner in which the information is transported from the producer to the consumer varies with the technology that is being used, which includes both online as well as offline communication. The diversity of accessible media makes it hard to rely only on a single system close to the source of a data file, to provide confidentiality, integrity, and availability. The security properties of a data file should follow the data file wherever it is used; not just be present at the initial moment when the file is released from the producer, but as long as the security policy of that file is valid.

Information systems are not only used by a single consumer retrieving information. Today, collaborations between organizations and sharing of digital resources are common. To achieve this and still maintain an adequate security posture, a highly flexible and adaptable security model is needed. The security model must allow for online and offline communication, execution and storage, but also the ability to share protected information outside the information owners' physical security domain.

1.1 Problem and research question

Object-Based Security (OBS)[1] is a vision of a security model which focuses on providing protection for a piece of information when stored or in transition by allowing for the security properties to follow the information. By attaching the security properties to the information itself, the access control decision is transferred from when the information is released from the original producer to when it is accessed by a consumer. Thus the information is not dependent on the original producer's security systems, yet can still maintain a security policy. How this kind of protection of the information is achieved is still an active field of research. Traditional approaches to access control range from system-oriented client – server relations to more data file-oriented cryptography solutions. In the client – server approach, the identity of the producer's system is verified and the communication between the producer and the consumer is protected by an encrypted tunnel. The trust relation in this approach is between the producer and the consumer – the consumer trusts that the producer is the assumed source and that the encrypted tunnel prohibits any outside interference. As the client – server

[1] Related approaches exist, for instance Object-Level Protection (OLP) and Content-Based Information Security (CBIS).

model does not explicitly protect the actual piece of information, it requires a direct path between the producer (or a trusted third party) and the consumer. Information obtained by other means, such as a portable memory like a thumb drive, cannot be verified as there is no connection to the producer. Likewise, the producer cannot establish any access control as soon as the information has left the controlled environment of the producer. An encrypted piece of information can maintain its confidentiality and access to the content is limited to those who have the proper decryption key. The efficiency and reliability of encryption lies in the type of encryption and the key management system. For the most part asymmetric encryption is used for key distribution and authentication. Thus, the consumer that receives the data file must be known at the time of the encryption, something that might be difficult in a cross-organizational collaboration. If a new entity (person or system) joins the group entitled to the information, the key has to be distributed or the information needs to be re-encrypted with a new key. Neither of these approaches is efficient for providing access control in a dynamic environment, where information is made available but where the consumers are either unknown or part of a group.

Attribute-Based Encryption (ABE) is a fairly new method of asymmetric encryption, introduced by Amit Sahai and Brent Waters (2005). In short, ABE incorporates an access policy in the encryption structure, thus combining encryption and fine-grained access control. The possibilities of fine-grained access control and the dissemination that encryption offers, makes ABE an interesting candidate to explore for the object-based security model.

The purpose of the task that is explored in this report is to investigate if ABE is a feasible candidate for providing access control to a piece of information according to the OBS vision, and how usable such a solution would be in a military environment. Specifically, this report shall answer the following questions:

1. In a military context, what are the benefits of using ABE in an OBS system?
2. How mature is ABE as an encryption method?
3. What would an OBS solution based on ABE look like?

Some applications of object-based security using attribute-based encryption will also be discussed although not in great detail.

1.2 Scope and delimitations

The purpose of this report is to present the possibility of using ABE for the OBS model.

As the focus is on applying the ideas from ABE as a proof of concept for OBS, the proofs of the completeness and trustworthiness of ABE as an encryption method are excluded from this report.

1.3 Research method

The work of gathering information was carried out as a literature study with focus on results from the research community.

1.3.1 Literature

An initial search for “attribute based encryption” in academic databases was conducted. These databases included Scopus and IEEE. As the concept of ABE was introduced in 2005 it was deemed that the total number of the result would be manageable. Thus, no restriction on age of the results was enforced.

The interest for attribute-based encryption has gradually increased over the years. A simple count using Google Scholar (Table 1) suggests the growing academic interest of ABE.

Table 1 Number of papers on "attribute based encryption" on Google Scholar.

Year	2005	2006	2007	2008	2009	2010	2011	2012	2013
No. of papers	6	9	45	90	119	192	363	581	825

Key qualifiers for candidate papers were descriptions and results providing knowledge of how ABE works, how mature the area is, and examples of applications of ABE.

A brief search was also conducted using the regular Google search engine. The purpose of that search was to identify commercial applications and initiatives. However, no commercial applications or initiatives of ABE where found.

1.3.2 Analysis

The first search in academic databases resulted in a set of about 200 papers. This set was reduced by studying the abstracts with the goal of identifying papers describing key concepts and examples of applications. As this report’s authors’ knowledge of the topic grew, further searches on specific subjects were conducted.

1.4 Terminology

Throughout the report the terms *producer* and *consumer* will be used. A producer is someone who creates or is responsible for an information object. It can also be the sender of a message. That is, the party that has something that someone else desires. The consumer is the one who requests and receives information.

The roles can shift depending on how a transaction is performed. Bob as a consumer may request information from Alice. Alice is then the producer of the information. However, if Carol asks Bob for the same information, then Bob becomes the producer and Carol is the consumer.

The purpose of the relation "producer sends something to the consumer" is to provide a consistent manner of describing communications. The terms come from the web service community. In other communities, these terms are sometimes described as sender – receiver or publisher – subscriber. Occasionally, producer and consumer are generalized as a *user*.

1.5 Outline of the report

The remaining part of this report is structured as follows.

Chapter 2 provides the reader with further background on access control and encryption, to provide context for the analysis and discussion.

Chapter 3 presents the results of the literature study of ABE, describing the function of ABE and current issues within the ABE research area.

Chapter 4 provides an analysis of why ABE is a suitable candidate to provide OBS functionality.

Chapter 5 discusses the results and analysis of Chapter 3 and 4 in the context of the Swedish Armed Forces.

2 Background

This chapter presents an introduction to access control and encryption.

2.1 Access control

Information systems usually contain a vast amount of information that should not necessarily be available for all users. To control which users can access what piece of information, an access control mechanism is needed.

Early on, access control was achieved through the use of an Access Control List (ACL). It is a simple but usable construction where the identity of authorised consumers is listed together with their assigned privileges, such as the right to read, write, and so on. The access rules were set and maintained by the producer of the individual resource, such as a file, while the consumer's identity was checked when accessing the network.

Although simple to create, ACL as an access control mechanism is cumbersome to maintain over time. As the file privileges are set by the producer, it is also the producer that is responsible for keeping the privileges current. If not, a consumer may accumulate access privileges over time, leaving the consumer with access to more information than is necessary for the consumer's current role.

A way to deal with the problem of identity-related privileges was the invention of Role-Based Access Control (RBAC). By adding the attribute role to an access policy, a slightly more fine-graded policy became possible. A consumer with the role project manager that changes position to manager may not be allowed to view documents from that project anymore. Already here, it is possible to identify where RBAC will fail or at least be inadequate. The role project manager is generic and applies to any project manager regardless which project the project manager is associated with. The same is true for enterprises that are located in different places. A system manager in location X, for instance, may not be allowed to manage systems in location Y. However, as far as RBAC is concerned, a system manager is a system manager regardless of location, and a project manager is a project manager regardless of which projects are being managed. For an RBAC system to work, complementary information is needed to differentiate between roles, such as project managers.

Attribute-Based Access Control (ABAC) is an access control mechanism which provides for a more detailed description of the consumer requesting a resource, but also for describing who is allowed to access the resource. ABAC uses, as the name indicates, a set of attributes to describe the properties of a consumer, the environment from where the request is made, and the properties of the resource. These three inputs are evaluated against an access policy at a Policy Decision

Point (PDP) before a Policy Enforcing Point (PEP) either grants or denies access to the resource. Figure 1 provides an overview of an ABAC system.

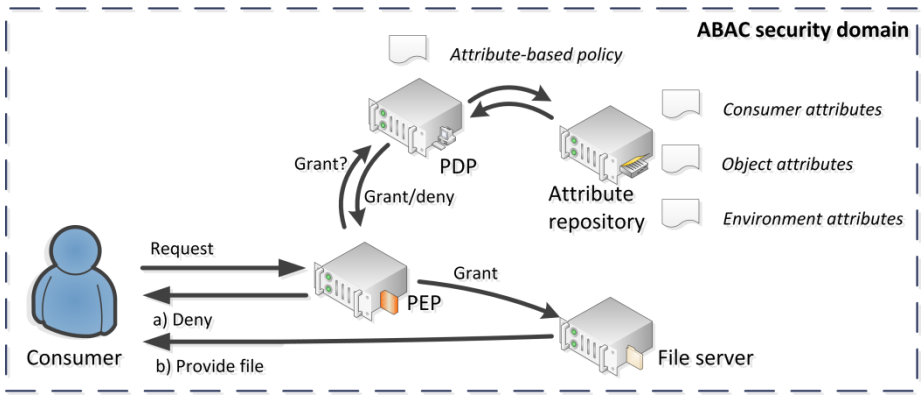

Figure 1 Overview of ABAC.

The trust model, i.e. the model of actors and actions that can affect the result of a security function, for ABAC is different from previous access control mechanisms. In an ACL system the producer of a resource sets the access rules based on identities. In RBAC the producer may choose which roles that can access the resource but the identities associated with that role are set by some administrative function. In ABAC there are several entities involved in the creation of access rules. The attributes describing the consumer are set by some administrative function responsible for maintaining personnel records. Environmental attributes describe the current situation in which the request is made. This could be the time of day the request is made, from which system the request is made, where the consumer is located, and so on. Object attributes are set by the producer of the object. These attributes should describe the content of the resource in such a manner that a policy can describe which user should be granted access under which environmental circumstances (Hu, Ferraiolo, Kuhn, Schnitzer, Sandlin, Miller & Scarfone 2014).

A narrow definition of an ABAC policy can be seen as an identity-based access control mechanism, like an ACL, with additional rules that further limit access. This however does not take the full potential of ABAC into consideration. A more flexible use of ABAC is to write policies that address properties of consumers, environments, and resources rather than identities.

2.2 Cryptology

This chapter provides an introduction to encryption. The purpose of the chapter is to introduce terms which are used later in the report. Readers familiar with symmetric, asymmetric and identity-based encryption may skip forward to the next chapter.

The basic idea of encrypting a message is to hide the meaning of the message from adversaries or illegitimate users by converting the plain text of the message to an unreadable and apparently random structure of characters (ciphertext). Historically, encryption was conducted using substitution of letters in the alphabet or with the use of a codebook. These methods where most often crude by nature but were supposed to work in the field where encryption and decryption were performed by hand. Modern cryptography is based on mathematical theory and, for all practical purposes, requires computers for encryption and decryption.

There are several more or less often used encryption schemes, but most of them can be categorized into two groups – symmetric and asymmetric encryption. A third category is Identity-Based Encryption (IBE). IBE uses the techniques from symmetric and asymmetric encryption but in a different manner. The following section presents an overview of these three categories.

2.2.1 Symmetric encryption

Assume that Alice wants to communicate privately with Bob, i.e. without any eavesdropper being able to understand what they are saying. To achieve this privacy, Alice takes her plaintext (*P*) message and encrypts it with a secret key (*K*). This results in a ciphertext (*C*) which is impossible to understand for an arbitrary listener. Bob, upon receiving the ciphertext, decrypts the message using a copy of the same key as Alice used, thus recovering the original plaintext. An encryption scheme that uses the same key for encryption and decryption is called a symmetric encryption scheme. An overview is shown in Figure 2.

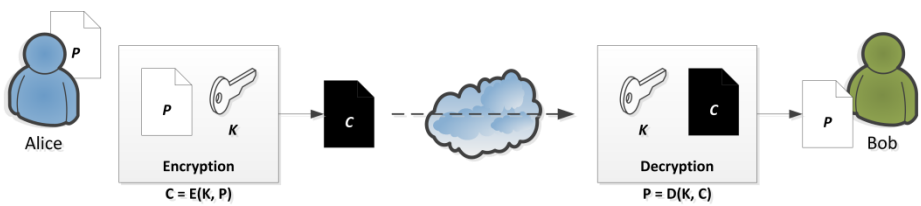

Figure 2 Alice encrypts a message and sends it to Bob.

Using the same key for encryption and decryption means that Alice and Bob must share a common key. If they want to include another friend (called Carol) in their conversation, Carol would also need a copy of the same key.

Symmetric encryption is efficient and considered more secure method than, for instance, asymmetric encryption. This is because a symmetric key has no dependencies, thus can be generated randomly. There are some drawbacks though. Since everyone participating in the conversation needs to share the same key, all keys must be replaced if one is exposed to a third party. Also, there is no secure way that Bob for instance can verify that a message is originally from Alice and not Carol as the message may have been encrypted by anyone in possession of the shared key.

2.2.2 Asymmetric encryption

Asymmetric encryption addresses the shortcomings of symmetric encryption by distinguishing between the encryption key (K_E) and the decryption key (K_D). These two keys are mathematically related to each other. That means that only the decryption key, which is generated together with the encryption key, can be used to decrypt a message (C) encrypted by the encryption key. Any other key will fail to decrypt the message.

Using the same example with Alice and Bob as before, Bob will calculate the two keys, and give one to Alice while keeping the other one. The key that Alice receives is Bob's public key, which is used to encrypt messages to Bob. The key that Bob keeps secret is the decryption key. Anyone in possession of Bob's public key can encrypt and send a message to Bob, but only Bob can decrypt a message encrypted with the encryption key corresponding to his decryption key. This scheme is more commonly known as public key cryptography and is depicted in Figure 3.

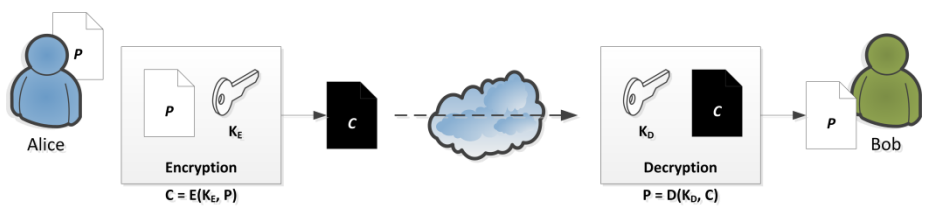

Figure 3 Alice encrypts a message to Bob using Bob's public key.

This scheme is one way, since Bob will need to obtain Alice's public key before replying to Alice.

Asymmetric cryptography is a very versatile tool within information technology. With public key methods it is possible to create certificates, i.e. assertions that a given public key belongs to whom the certificate claims.

Another use of asymmetric encryption is for signing messages. This is done by reversing the scheme. Bob signs a public message using his secret decryption key. Now anyone in possession of Bob's public encryption key can decrypt the string and compare it with the public string. If there is a match then Bob (or at least his secret key) must indeed have been the one to sign the message.

In most cases, an infrastructure is formed for handling certificates. This is called a public key infrastructure (PKI). By allowing for a trusted third party to perform the generation of key-pairs and signing of certificates, a receiver does not need to know or trust the individual behind a certificate beforehand.

The downside of public key cryptography is that it is a fairly slow process. Compared to symmetric encryption, public key encryption typically requires more time-consuming calculations. This does not lend asymmetric methods well to bulk encryption, for instance large or streamed files. However, the ability to validate the origin of a sender has lead asymmetric methods to be dominant in the set-up of communications by providing authentication and key exchange. As a part of the set-up, when origin has been established, a symmetric key is usually transferred for more efficient communication during the rest of the session.

2.2.3 Identity-Based Encryption

Identity-Based Encryption (IBE) is a variant of asymmetric encryption, where the public key of a user contains some piece of information already publicly available but still uniquely related to the user. This could for example be an e-mail address or a username.

IBE is not without its limitations and disadvantages. However, these are not considered in this section as they will be discussed in conjunction with ABE.

Once again, the scenario of Alice and Bob communicating by sending each other messages is used to illustrate how IBE works, as shown in Figure 4. The centre of operation in IBE is the Private Key Generator (PKG), which holds a public key-pair: the Master Public Key (P_{PKG}) and the Master Secret Key (S_{PKG}). When Alice wants to send a message to Bob for the first time, she fetches the Master Public Key from the PKG, and uses it in combination with the publicly available identity information related to Bob to encrypt the message. To decrypt the message from Alice, Bob needs to fetch the Master Public Key from the PKG and acquire his secret key which corresponds to the public key Alice created. Bob retrieves the secret key (S_{Bob}) from the PKG and can then proceed to decrypt Alice's message. In the future, Bob can use the same key when reading messages encrypted with his identity. The same is true for the Master Public Key.

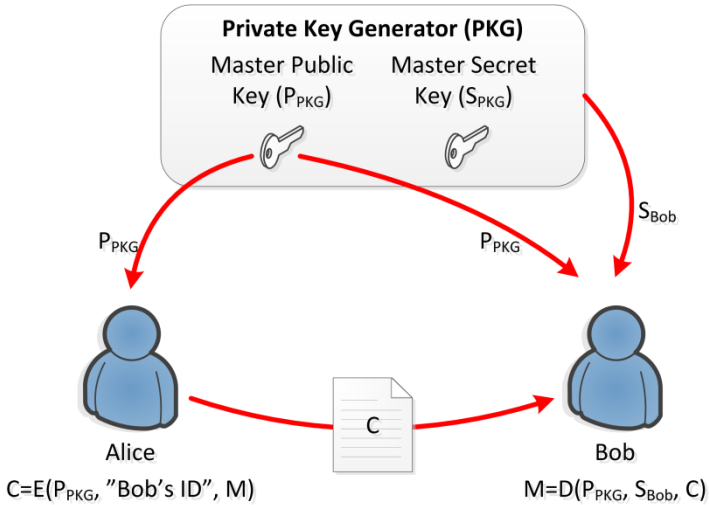

Figure 4 Overview of IBE.

IBE simplifies the key distribution process of asymmetric encryption, since there is no need to pre-distribute keys between communicating individuals and thereby reducing the need of supporting infrastructure. There is no need for a PKI, whose purpose is to distribute public keys, with the associated considerable management, as the keys are generated when needed..

Neither is there a need of managing certificate revocation lists (CRL). A CRL is a database containing revoked certificates, that is, certificates which are no longer valid, due to private keys being lost or stolen, policies violated or keys that in any other way are deemed compromised. Revocation is different from expiration; revocation handles the situation where unexpired certificates due to the above mentioned reasons are no longer valid, while expiration concerns a pre-set time limit. CRLs need to be available all the time, either for download or as an online service and they are expensive to implement, maintain and manage.

The PKG generates secret keys for any participant, and is thereby capable of decrypting any message. This does generate considerable security requirements on the PKG to make it possible for Alice and Bob to trust it. In a way, this also applies to key generating authorities that generates symmetric or asymmetric keys. The key generation is one of the most sensitive operations within encryption.

The concept of IBE has been further developed into the concept of fuzzy IBE, where biometric information of users constitutes the unique user information. Since measurements of biometric attributes may differ slightly each time, fuzzy

IBE is error-tolerant, thus allowing for more valid values of identity. In the next development step, fuzzy IBE was generalized into (ABE), by incorporating more attributes about the user and an access control policy based on the attributes. The attributes of ABE may for example contain organizational affiliations. ABE will be detailed in chapter 3.

3 Attribute-Based Encryption

This chapter provides an introduction to the research on Attribute-Based Encryption (ABE). Although the topics presented give a broad view of the current status of research activities, the selection is slightly limited to reflect the issues that are relevant for Object-Based Security (OBS).

3.1 Key concepts of ABE

ABE has its origin in Identity-Based Encryption (IBE), wherein the use of the consumer's identity is used to create an asymmetric key-pair. As IBE evolved, ABE was invented by generalizing the consumer's identity to a broader set of attributes that would describe the intended recipient of a message. By using a set of attributes to describe the consumer, it is also possible to create a policy of how these attributes should relate to each other in order to grant the consumer access to the file in question. Thus, ABE provides asymmetric encryption that is combined with descriptive attributes and a policy that provides access rules based on a set of attributes.

3.1.1 Master Key Generator

The generic structure of ABE is similar to IBE. This means that the producer of an encrypted message does not need to obtain a key associated with the consumer prior to encrypting and sending a message. Furthermore, as ABE uses descriptive attributes instead of an identity, the sender is encrypting a message which can be read by anyone who satisfies the policy associated with the attributes. Thus, ABE messages are directed to groups rather than to individuals.

The centre of operations is the Master Key Generator (MKG), which corresponds to the Private Key Generator (PKG) in IBE. When the MKG is set up, it generates a Master Secret Key and a public key. The Master Secret Key is the master secret within ABE, and it is with the master key that decryption keys are generated. The public key provides the producer with the necessary input for encrypting a message under a set of attributes or an access policy.

A consequence of having a master key is that ABE will have inherent key escrow, as decryption keys are fetched as needed by authenticated consumers. Key escrow is common in public key infrastructures (PKI) as well, but with the difference that the PKI key escrow is voluntary and only holds the keys put there. The MKG can generate any decryption key necessary for messages encrypted under the public key.

3.1.2 Secret sharing schemes

Throughout the years there have been different flavours of ABE. In the initial proposal (Sahai & Waters 2005) there was no policy, but only a set of attributes which needed to be satisfied. This view originates from the secret sharing schemes. The basic idea of secret sharing is that *k*-out-of-*l* shares of information are needed to recover a secret (Simmons ed. 1992). That is, with a total of *l* private pieces of information – or shares – *k* unique shares are needed to recover the secret. This constitutes a situation where the cryptographic key is a composite of *k* shares. Thus, cooperation – or collusion as described by Simmons (red. 1992) – between *k* participants (or insiders) is needed to recover the secret. An attempt to recover the secret with less than *k* shares leaves the secret completely undetermined; that is, all values of the secret are equally likely and the secret cannot be recovered.

Secret sharing has advantages in settings where dual control is necessary to perform an operation, such as missile launches, or where several parties are involved, such as when updating keys for signing the Domain Name System (DNS) in DNSSec for the whole Internet (Internet Corporation for Assigned Names and Numbers (ICANN) 2014). The main idea in secret sharing is to obtain robust key management, where an operation should be possible to carry out even if all legitimate key holders are not available. For instance, assume that an operation needs two keys to be allowed to be executed. A group of five members may each have a key which, when combined with another arbitrary member's key, will be able to execute the operation. Thus, the ability to execute the operation is not solely dependent on two specific individuals, rather two out of a group of five. This provides a higher availability of the operation and, at the same time, prohibits a single member to perform the operation by themselves.

ABE is used to encrypt messages directed toward a group of individuals. However, the participants in different groups should not be able to collaborate in order to gain more privileges than intended .thus, ABE is reminiscent of secret sharing due to different pieces of information being needed to decrypt, but with the important difference that whereas secret sharing is based on cooperation between different participants, cooperation is expressly forbidden in ABE. The idea of ABE is that the different pieces of information are attributes related to the same individual (Goyal, Pandey, Sahai & Waters 2006). Such attributes may be personal characteristics, such as biometric features, but it may also be occupational traits, such as organizational position and responsibilities.

3.1.3 Key-Policy Attribute-Based Encryption

Goyal et al. (2006) improved ABE by providing an access structure based on the attributes, thus making the access control more fine-grained. The access structure (or access policy) can be depicted as a tree structure where leafs represent the

attributes and the connecting nodes are either AND, OR, or threshold gates. With the access structure, collusion resistance was also achieved.

The main idea of Key-Policy ABE (KP-ABE) is presented in Figure 5, which is a simplified version of (Wang, Zhang, Schooler & Ion 2014). Alice wishes to produce an encrypted document. To do so, she requests the public key of the MKG and encrypts it, using a set of attributes that describes the document. She can then publish it, along with the attribute set. Bob, who is one of the consumers of Alice's document, retrieves the encrypted document and asks the MKG for the public key and his private key. Bob's private key includes an access policy which regulates what Bob is allowed to view, based on a set of attributes in the policy. If the policy in Bob's private key is satisfied by the attributes provided in the message from Alice, he can decrypt the document.

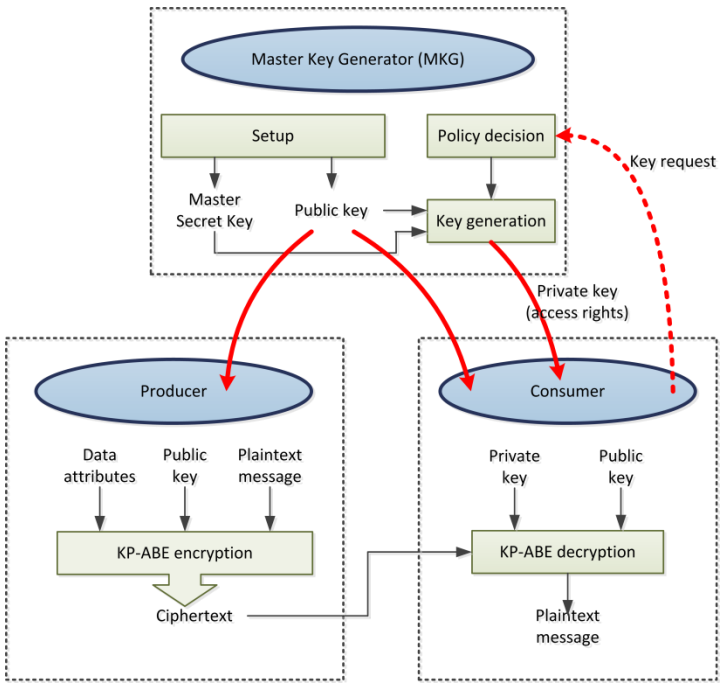

Figure 5 Key-policy attribute-based encryption.

The setup in Figure 5 is generic. It is often suggested to encrypt the message using a symmetric key and then encrypt the symmetric key using ABE. Appendix A: Key-Policy ABE provides a description of KP-ABE with the use of Advanced Encryption Standard (AES) for message encryption.

By encrypting a document, Alice, if she wishes, can publish the document on a third party storage area, thus making the document more available for the intended recipients. Once Bob has retrieved his private key, he can decrypt any message encrypted with the public key as long as his policy is satisfied. Thus, no communication with the MKG is necessary after the initial key retrieval.

There have been a few suggestions for the possible application of KP-ABE, such as targeted broadcast and audit log analysis (Goyal et al. 2006) but also secure cloud storage (Yu 2010). A content provider can choose to encrypt its content and provide it over a broadcast channel. A recipient will receive a decryption key (the private key) from the content provider containing a policy that will give access to the content the content provider and receiver have agreed upon. Another usage is for providing restricted access to an audit log. The full content of an audit log may be considered sensitive information, thus a key with a policy regulating which part of the audit log the analyst is allowed to see will limit the ability to get full knowledge of the content.

3.1.4 Ciphertext-Policy Attribute-Based Encryption

Ciphertext-Policy ABE (CP-ABE), introduced in (Bethencourt, Sahai & Waters 2007), reverses the way that KP-ABE operates. In CP-ABE, Alice attaches an access structure to the document before it is published. As a consequence, Bob's private key consists of a set of attributes that describes Bob's current situation regarding for instance organizational position, associations, or status.

The main idea of CP-ABE is presented in Figure 6, which is a simplified version of (Wang et al. 2014). Alice wants to encrypt a document. She retrieves the public key from the MKG and uses it together with an access structure she has created based on some access control policy, to encrypt the document. She can then post the document on a storage area of her own choosing. Bob fetches his private key, containing his descriptive attributes, along with the public key from the MKG. If the attributes in the private key matches the attribute in the access structure in a correct manner, Bob can proceed to decrypt the document.

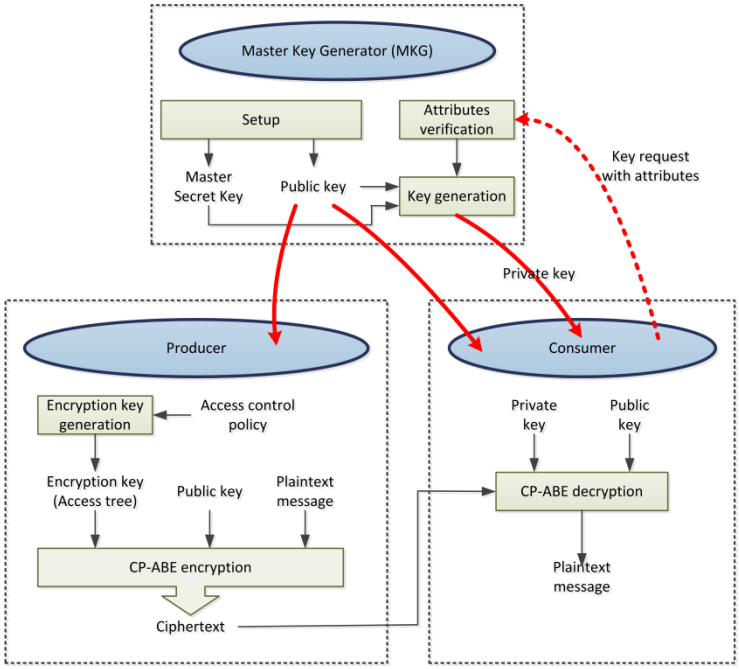

Figure 6 Ciphertext-policy attribute-based encryption.

The setup in Figure 6 is generic. It is often suggested to encrypt the message using a symmetric key and then encrypt the symmetric key using ABE. Appendix B: Ciphertext-policy ABE provides a description of CP-ABE with the use of AES for message encryption.

In addition to further developing fine-grained access control, the main contribution of CP-ABE is to give control over access rights to the producer of the ciphertext. This is strictly different from the scenario of KP-ABE where the MKG has control over access rights.

Within the literature there are more examples of CP-ABE being implemented compared to KP-ABE. Typical examples are securing medical records (Akinyele, Pagano, Green, Lehmann, Peterson & Rubin 2011) and cloud storage solutions (Yu 2010). Common for both examples is that they want to provide a higher availability of information. In the cloud example, storing information on third party servers is a way to make documents more available and still restricting access for those who do not satisfy the policy. In the medical records example, authorized users may access certain part of the record depending on their needs, i.e. attributes.

3.2 Attributes and policies

Although ABE initially had a *k*-out-of-*l* threshold policy, this was quickly replaced with a tree structure to provide a more fine-grained access control. Goyal et al. (2006) suggested a monotonic access structure, which means an access structure containing attributes in the leafs, and AND, OR, or thresholds operators in the intermediate nodes, see Figure 7. Other operators, such as greater than or less than, can also be used for more fine-grained access structures.

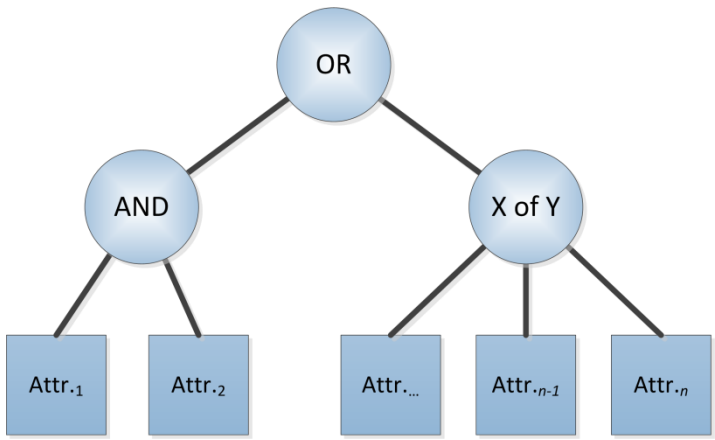

Figure 7 Monotonic access structure.

A monotonic access structure is defined as one where any subset of the attributes is a part of the access structure. A secret sharing scheme is a typical monotonic access structure, but it also works for tree structures as illustrated in Figure 7.

A development of the monotonic access structure is the non-monotonic access structure. In a non-monotonic access structure, attributes which should not be present for access to be granted, are displayed as negative attributes or NOT-attributes. The non-monotonic access structure works with the tree structures of KP- and CP-ABE. As the negation of an attribute does not correspond to the logical NOT-function, rather it is a complementary attribute, it is natural that the attribute-base becomes twice as large. Ostrovsky, Sahai and Waters (2007) suggested a solution where the negative value is generated by keys, thus avoiding doubling the number of attributes.

The receiver of a message under a CP-ABE policy retrieves a secret key from the MKG. This key contains the attributes that describes the current view of the user. However, if the situation changes and a value of an attribute need to be changed, the whole key is replaced with a new key that has the updated attribute. Generating secret keys could be a costly process if it happens frequently and

especially on a system with many users. Chen and Gerla (2009) have devised a method where attributes can be updated without regenerating the whole secret key. They call such attributes dynamic attributes.

In an ABE system for an enterprise, attributes needs to be controlled. The purpose of the control is to provide a closed set of attributes. If not, attributes can be randomly chosen both for encryption and decryption, making it impossible to define policies that apply to a group. Using many attributes in a policy tree causes a lot of overhead. However, it is not uncommon that there is an implicit hierarchy among attributes. Li, Wang, Wang and Ren (2009) proposed Hierarchical ABE (HABE) to lessen the overhead. This is achieved by ordering the attributes in a tree structure where a sibling of an attribute can be derived from the parent but not the opposite. Thus a parent can be revoked and as a consequence all children of that parent (node) will be revoked as well. The rest of the tree structure will remain intact.

For most applications of ABE, the policy and the attributes are not encrypted. However, there are situations where this may be preferable. Depending on the reason behind the policy, the available attributes and policies may reveal information about the sender or receiver that they do not want to disclose. Kapadia, Tsang and Smith (2007) propose a system where the policy is hidden so that the receiver of an ABE encrypted message cannot learn how the policy is structured. Thus the intention behind setting the policy remains a secret. Yu, Ren and Lou (2008) suggest a hidden policy application for content delivery networks where the purpose of hiding the policy is to protect business competiveness and user information privacy.

Dual policy, as described by Attrapadung and Imai (2010), combines KP-ABE and CP-ABE. Each of the policy methods has their merits and drawbacks, but they can complement each other. For instance, an information owner may want to control who gets to decrypt a certain piece of provided content. On the other hand, there are times where the content owner may want to update or remove specific attributes from a user. Both these situations are possible in KP-ABE and in CP-ABE, respectively. However, for full control the content owner may want to use both methods.

3.3 Performance

Adding a security mechanism like cryptography will always result in some performance degradation compared to just sharing a file as an attachment in an e-mail. Commonly, symmetric algorithms like the Advanced Encryption Standard (AES) are used for bulk encryption due to the efficiency and speed of the encryption and decryption operations. Asymmetric encryption like Rivest-Shamir-Adleman (RSA) is usually a slower method for bulk encryption but, on the plus side, allows for the encrypted file only to be opened by a user with the

corresponding key. As ABE is an asymmetric method, it maintains the ability to choose recipients. However, ABE addresses recipients as a group and access to this group is managed through a policy. The cost of having a policy as evaluation criteria for decryption comes with the price of being slower than RSA and AES. However, neither RSA nor AES provide any fine-grained access control.

In ABE, the access structure and attributes are embedded in a message. The size of the encrypted file grows linearly with the amount of attributes used and, consequently, the size of the access structure (Hohenberger & Waters 2013; Waters 2011). When it comes to execution time however, the amount of attributes has less impact on the end user's device. Having too many attributes is probably infeasible as it would be difficult to write a usable policy. With less than ten attributes, the time it takes to encrypt a file is a fraction of a second on a laptop, something the user will hardly notice. What really affects the time to encrypt and decrypt messages is the length of the symmetric key (Wang et al. 2014).

The complexity of the access structure or policy also affects the time it takes to encrypt and decrypt files. A simple monotonic structure with a threshold policy is less complex than a monotonic tree structure. Enhancing the monotonic access structure to a non-monotonic structure by adding NOT-attributes, not only increases complexity (Qiao, Liang, Davis & Jiang 2014), but it also doubles the amount of attributes in the attributes space.

In more generic terms there is also a conceptual difference between KP-ABE and CP-ABE. These results in a noticeable difference in execution time, where KP-ABE is faster than CP-ABE in all operations due to the fact that CP-ABE needs to make more exponentiation operations, for instance when generating the access tree (Wang et al. 2014).

Wang et al. (2014) also compared the time it took to decrypt messages on a laptop and a mobile phone. The laptop has enough processing power to be able to decrypt files encrypted with an 80-bit symmetric key and ABE with 30 attributes in under a second. The mobile phone, however, was over ten times slower making it an infeasible platform for ABE at the moment. Traynor, Butler, Enck & McDaniel (2008) managed to get better performance on a mobile phone for their IPTV-broadcast system with a few seconds for decryption. In their experiment however, they used a threshold policy which is less complex than tree structures.

3.4 Revocation

A change in policy will prompt the need to withdraw files encrypted under the old policy or to reissue private keys with the new policy included. A change of attributes will prompt the same need for withdrawal of old files. The reason

behind the policy or attribute change can be due to new business relations or structural changes within the organization. Another reason can be that a user changes position within the organization and thus will change the needs of consuming information. There is a natural limitation of withdrawing already released information, especially if the information is released outside the producer's security domain. Revocation is a common term in PKI, relating to the need to render an active key-pair invalid before the public key certificate expires. The motivation for revoking keys can be that the trust in a public key has been lost for some reason most likely due to that the private key has been compromised somehow. In ABE, the term revocation is extended in meaning as it involves keys, rather than certificates as PKI does. Another thing to keep in mind is that PKI keys are personal, thus are not changed if the holder of the keys, for instance, changes position. ABE keys are bound to change over time as they represent the current view of files and users. Thus, revocation in ABE means the withdrawal of permissions issued to a user and is conducted in one or more of the following ways:

- Changing the policy.
- Withdrawing the private key.
- Leting the private key expire.
- Leting the private key attributes expire.

There has been a variety of proposed solutions to the revocation problem, see for instance (Ming, Fan, Jing-Li & Zhao-Li 2011; Yu, Wang, Ren & Lou 2010; Wang, Liu, Wu & Guo 2011), but so far not a single solution that handles all the suggested situations has been proposed. For example, in an offline scenario a private key cannot be automatically withdrawn, it has to expire. This does not suggest that it is impossible to build useful solutions to deal with revocation but depending on the requirements there will be drawbacks. Still, the most reliable methods of revocation in the ABE context, is to let the key or the key attributes expire. It is possible to try to withdraw a user's private key or information objects but this method cannot guarantee success in doing so. Changing policies has the effect that new information objects cannot be read by consumers with revoked keys but old information objects are still readable, at least until the key or one of its attributes expires.

Changing the policy may force a re-encryption of stored files, so that they comply with the new policy. In deployed systems with a huge number of files stored on servers, the computational load for re-encrypting files could be extremely high. Files not stored on the servers will of course not be re-encrypted. By applying proxy re-encryption and lazy re-encryption into the ABE scheme (Wang et al. 2010) the computational load will probably be significantly smaller. A proxy re-encryption service is a way to offload the MKG by sharing some of the burden of encryption. Lazy re-encryption means that a file will be re-

encrypted when someone asks for it and not immediately when the policy is changed. Another approach that has been suggested, to have the means to directly revoke a user's key, is to employ an online mediator that must be involved in every decryption (Yu et al. 2010). This mediator will not cooperate if the consumer's key has been revoked. This is a combination of ABE and classic access control systems. Since the mediator must be online at all times in order for consumer's to be able to decrypt files this imposes limitations on the availability of the system and it alienates offline users.

Dealing with revocation will fundamentally involve trade-offs that must be evaluated for each environment. Any implementation that addresses this issue must take both online and offline approaches into consideration.

3.5 Expanding the applicability of the MKG

Building a basic system designed for the use of ABE requires only a few components. These components are two software programs: the MKG service and a program to encrypt and decrypt files. The MKG service preferably runs on well-protected server hardware to ensure that unauthorized persons do not gain access. Software for encryption and decryption runs on the users' own computers or mobile devices.

The MKG service creates keys used to encrypt and decrypt messages. A great advantage is that this service need not necessarily be online for ABE to work. Once keys are created and distributed, there is no need for user interaction with the MKG until new keys are needed. Thus, the MKG can achieve a stronger protection by being less exposed. With an offline solution however, there is a need for a manual distribution plan for keys, especially the MKG's public key.

Decentralized or Multi-Authority ABE is an interesting topic introduced by Chase (2007) and further developed by, among others, Chase and Chow (2009) and Lewko and Waters (2011). In almost all ABE proposals, private keys are issued by a single central authority. In a Multi-Authority environment, attributes originate from different authorities, and need to be gathered to form a private key. In a single authority system, some attributes may suffer in correctness if the authority is not responsible for the maintenance of the attributes. Collecting attributes from authorities with maintenance responsibility results in a more accurate description of user attributes. An example could be nations participating in a coalition; each nation is responsible for their user attributes but there may be a need to provide a coalition-wide key. Thus, that key is generated with attributes from several authorities within the coalition.

In many applications a party will want to share data according to a policy written over attributes or credentials issued across different trust domains and organizations. In a commercial application, two corporations might both issue

attributes as part of a joint project. Using normal centralized ABE systems for these applications can be problematic as a single authority needs to be able to verify attributes across different organizations and issue private keys to every consumer in the system.

One major challenge with Multi-Authority ABE is to prevent collusion, as a consumer could collect different keys from multiple authorities, thus being able to combine keys to decrypt messages the consumer is not entitled to. This problem is given a solution in (Chase 2007) by linking each user through a Global Identifier (GID) to the user's keys from different attribute authorities. Using a GID though, does provide a corrupt authority with the ability to trace users. Several contributions to prevent this problem have been summarized by Pang, Yang and Jiang (2014) in a survey on multi-authority ABE.

Scalability is a current challenge in developing effective ABE systems (Qiao et al. 2014). A way to minimize the risk of having a single MKG in a growing security domain is to use hierarchical MKG's. A Master Secret Key and a public key are generated for each subdomain MKG, thus spreading the risk within the MKG's for collusion and loss of confidentiality (Wang et al. 2011).

3.6 Experimental applications of ABE

Although ABE cannot yet be found as a commercial product, there exist some experimental implementations and applications. Early suggestions of usage of ABE comprised of file storage on third party servers, protection from insiders, restricted access for log analysis, and conditional access to a broadcast flow.

In the case of audit logs, analysts are usually given full access to the entire log, something which could become a security risk if the analyst goes rogue. Goyal et al. (2006) suggested using KP-ABE to provide an access policy, thus only making a portion of the audit log available for each analyst. The policy can, for instance, only make events regarding a certain period of time visible, or events relating to a specific user identity.

Another suggestion, also by Goyal et al. (2006), was to use KP-ABE to provide conditional access to broadcast television. An access structure was used to decide which packets to decrypt in a stream of packets originating from the provider. Traynor et al. (2008) made a large scale experiment of this for a system with over 26 million viewers. In their experiment the access structure was simplified to a threshold policy but with a large set of attributes (up to 100 000 attributes).

There have been several papers and theses written about storage of ABE encrypted files on untrusted storage areas, for instance by Zhu, Yang and Wu (2013) and Yu (2010). The main issues surrounding third party storage has not been around the actual encryption as ABE utilizes an underlying encryption method like AES. Rather, the discussions focus around the privacy issues that

arise from the fact that attributes and the policy are often visible for anyone who can access the related encrypted file.

Arguably the largest area of interest is that of public health records. In this area, several of the central ideas behind ABE are being explored. There are several users who during certain conditions are allowed to view the information in a record. Typical for the health community is that the only constant is the patient, whereas doctors and nurses may change from time to time and with location. Critical issues for public health records are how restricted access to information in the record can be achieved and under which circumstances the information can be accessed. Some notable references in this area are Akinyele et al. (2011) and Li, Yu, Zheng, Ren & Lou (2013).

4 Analysis

In this chapter, the properties of Attribute-Based Encryption (ABE), as documented in Chapter 3 are analysed in relation to the needs of Object-Based Security (OBS). The three research questions stated in Section 1.1 sets the frame within which this analysis is conducted:

1. What are the benefits of using ABE in an OBS system?
2. How mature is ABE as an encryption method?
3. What would an OBS solution based on ABE look like?

The answers to these research questions indicate to what degree ABE may provide OBS functionality.

4.1 General observations regarding ABE in relation to OBS

The basic ideas that are the foundation of OBS are the ability to statically protect information without the support of other IT systems and to enforce access control to that information.

Static protection is typically enforced using cryptography when information resides outside the security domain of the producer. However, common encryption techniques do not themselves provide access control functionality to the information it protects. For systems which would use a symmetric encryption method, all members are entrusted as it is the same key used for encryption and decryption of a message. This makes a symmetric key a group key. By using asymmetric encryption methods, encrypted files can be tied to a single receiver as the encryption key only corresponds to a single decryption key.

In this report, thus far, access control has been enforced through the possession of a key which can decrypt a given message. Having possession of a key (hopefully) means that the holder of the key was entitled to the information at the time that the key was provided. If the privileges of the holder have changed since then, the key has to be retrieved or the information needs to be re-encrypted. Usually, access control is enforced when a consumer requests a new private key.

ABE provides several advantages over traditional encryption methods due to the included policy. This policy allows for the producer to set a policy which grants access to a group of consumers rather than a set of individuals. Availability, a property often less focused on in security discussions, has through the development of ABE gained more focus.

As ABE is a fairly new and active research topic, there are several variants of how to implement different functionality. In this analysis a broad view of ABE is presented by not referring to specific schemes other than in the references.

4.2 What are the benefits of using ABE in an OBS system?

In this section some benefits of using ABE in an OBS system are identified and analysed.

4.2.1 Third party storage properties

In collaborations between two or more organizations, information needs to be exchanged between different security domains. Granting access, with specified restrictions, to groups or roles is interesting in such scenarios. ABE research has also developed schemes for decentralized and multi-authority access control. This method support collaborations and coalitions, where the servers of the collaborating parties may be considered as third party storage.

The ability to store information in a third party storage facility was one of the earlier challenges for ABE researchers. The growing trend of storing data, even personal or otherwise sensitive data, on third party services increases the risk of these data storage sites being attacked and data compromised. Examples of such services is different cloud services, Google et cetera. In ABE, only consumers with a certain accepted set of attributes may decrypt a document, which makes it possible to store sensitive documents on untrusted storage servers instead of on trusted servers with capability of authenticating users before allowing them access to the documents.

The concept of fine-grained access control based on sets of attributes is a main quality of, and contribution from, the ABE development. Combining attributes needed to obtain access to information can be achieved through the use of, but not limited to, the logical functions AND, OR and NOT. It is also possible to state that *k* out of *l* different attributes are required in a threshold function.

The concept of fine-grained access through ABE may also be applied in the setting of targeted broadcast, where for instance access may be given to certain TV-series or certain seasons of TV-series, according to the subscription that the consumer have bought. Targeted broadcast may be used in other settings of widely distributing information with restrictions in access to the information. A possible military scenario regarding broadcast is the information flow that provides a Common Operational Picture (COP) during an operation. With a policy regulating which information is available to whom, a selected view of the

situation can be achieved. Thus, a COP can be distributed on a need-to-know basis, within one force or between forces in a coalition.

4.2.2 Offline properties

Basing access control decisions on ABE opens up to being able to cope with situations where users are isolated or partly isolated from the Internet or other infrastructure. Such situations are plausible in for example different military scenarios, where external communications to the Internet or Armed Forces infrastructures may temporarily have been lost or otherwise unavailable. Operative decisions need to be made even in those situations and available information will be the basis for these decisions. It is therefore critical that the best possible information is available.

The handling of medical and patient records is a vivid focus area in current ABE research efforts. This is probably partly due to ABE being considered to be an interesting approach to handling privacy sensitive data in offline settings. Experiences from this area ought to be valuable to Armed Forces scenarios, even though privacy issues are probably not the main focus in those settings. Offline settings may occur for example due to situations of crisis or catastrophes, such as the hurricane Katrina (Akinyele et al. 2011). During the hurricane Katrina access to medical records were limited or non-existent in several situations, even at permanent health care facilities. A possible solution or at least a solution that would have improved the situation would be if people could carry a copy of their own medical record. ABE could be used to protect those records with an access policy, thus regulating access even in an offline scenario.

The discussion above indicates that communicating partners may occasionally find themselves in an offline situation. This is also true for the infrastructure or parts of it, such as the MKG. In fact, being offline at certain points or most of the time may even be desirable (Wang et al. 2011; Yu et al. 2010).

As a central resource for the managing of keys, the MKG is critical. An important question is how often a new key is needed and, due to this, how often communication with the MKG will be required. The answer to this question probably depends on technical choices and solutions as well as organizational and other requirements from the practical settings or environment in which the ABE solution is implemented. The dynamics and requirements of the individual environment will influence how often attributes will need to change and thereby also how often new keys are needed. ABE is basically an encryption method, but with the capability of enforcing fine-grained access control through the use of attributes. Since ABE thereby is not a client – server based solution, an offline situation is well within the realm of ABE's capabilities.

4.2.3 Policy properties

Within the OBS vision, communication patterns and access control needs are highly dynamic. Information in motion is a highly simplified way of describing the typical OBS scenario. This does however put tough requirements on security and creates a need to find a balance between confidentiality and availability requirements. In offline scenarios, for instance, this balance is delicate.

A major advantage of ABE related to the OBS vision is the fact that ABE provides group keys, facilitating communication within groups and not primarily between individuals. Which individual who decrypts a message is not of primary concern in an ABE setting. Instead, decryption is allowed by anyone fulfilling a policy based on attributes and the relation between these.

ABE research has so far made efforts to accommodate a fairly broad spectrum of needs and requirements on policies and the attributes on which policies are based and designed. The different steps taken from the simpler k-out-of-l threshold policies to more sophisticated monotonic and non-monotonic access structures facilitate describing more complex and realistic organizational and operational settings. More complex access structures come with a computational cost, which is especially critical when using simpler mobile units. The computational cost may be related to generating updated secret keys caused by updated attribute values, handling of large sets of attributes, size and complexity of the policy, key revocation and re-encryption due to policy changes.

Efforts within current ABE research intend to lower the computational cost, for example in the case of handling large sets of attributes by introducing Hierarchical ABE (HABE). The introduction of dual policies, combining the advantages of KP-ABE and CP-ABE regarding control of who may decrypt a message and control of attributes related to specific users, is another type of improvement of the ABE concept. From an OBS point of view it is critical that the policy offers adequate access control and that the computational cost is acceptable, even when using mobile devices offline.

Handling privacy issues of individuals tends to be focused on the information itself and what it states about the individual, especially regarding more sensitive information. Hiding policies may contribute to providing privacy in this sense. Furthermore, hiding policies may provide protection of an organization’s prioritization and assessment of information as disclosed by a policy. The latter may be of certain interest related to the needs of the Armed Forces, for example in settings of coalitions and collaborations.

What techniques and methods needed to create suitable policies has not been a primary ABE issue, at least not within the research community. However, to make OBS usable, some thoughts on usability and user friendliness may be needed. Practical in-field experience will to a large degree be needed, in conjunction with research results to create policies which are both effective in an

operational sense and related to the current status of the operative environment. Predefined policies and standard sets of attributes adjusted to different activities and operations may be one way of simplifying the management and general handling of policies. The development of policy trees or access structures incorporates mechanisms able to describe complex relationships, which when systematically used have a potential for describing the status of the operative environment. This might for example be in the sense of hierarchies among attributes, logical relations between attributes and whether policies ought to be hidden.

The human difficulty of classifying and deciding who should be given access to information is a major challenge. ABE provides possible means and measures to describe complex access control scenarios, but to what degree ABE solutions will be successful in operative conditions is also related to man – machine and organizational issues. The operative environment will probably be more complex than what can be modelled by a policy at an acceptable computational and economical cost and in a way which provides usability, confidentiality and information availability. To find the adequate balance between these different qualities, the perspectives of different information consumers in an OBS environment will be needed.

4.3 How mature is ABE as an encryption method?

ABE is currently an active area of academic research, where different ideas and strategies are tested to see what may work. No commercial ABE development efforts seem to exist so far. Thereby, ABE currently cannot be considered a fully mature technology, but there are active research efforts to further exploit ABE's potential of handling real-life needs and challenges.

The different ABE research efforts have an interesting potential to facilitate sending and storing information outside the normal security domain of the producer, providing trusted offline authorization and a more fine-grained access control. These are qualities which make this research area interesting in the context of OBS. Sending and storing information outside the producer's security domain requires development of policies adjusted to a broad spectrum of activities and operations.

The whole family of ABE research is a fairly new direction within cryptographic research. There is a vivid dialogue and information interchange going on between what researchers discover is possible to do with ABE versus the expectations on how ABE may contribute to fulfilling needs and requirements in operative environments.

There are two main directions of ABE research, KP-ABE and CP-ABE. The development from the earlier efforts resulting in KP-ABE mutated further into the CP-ABE track of research. Currently the CP-ABE track of research seems to be dominating over KP-ABE. ABE research has further developed into sub-branches facilitating multi-authority and hierarchical access control. This may also be interpreted as a part of the dialogue and interchange between research and expectations on ABE, resulting in policies for more complex and potentially more realistic real-life settings.

Table 2 TRL-levels (EDA n.d.).

TRL	Definition
1	Basic principles observed and reported.
2	Technology concept and/or application formulated.
3	Analytical and experimental critical function and/or characteristic proof of concept.
4	Technology component and/or basic technology subsystem validation in laboratory environment.
5	Technology component and/or basic technology subsystem validation in relevant environment.
6	Technology system / subsystem model or prototype demonstration in a relevant environment.
7	Technology prototype demonstration in an operational environment.
8	Actual technology system completed and qualified through test and demonstration.
9	Actual technology system qualified through successful mission operations.

Technology Readiness Level (TRL) is one measure used to assess the maturity of technologies. There are several existing definitions of TRL, with varying numbers of levels (7–9 levels) and slightly diverging definitions of each level. Still, the overall idea of all definitions of TRL is to describe the transition from conceptual research and development ideas to fully mature technology in operative environments. The TRL, as defined by The European Defence Agency (EDA) (n.d.), are presented in Table 2.

TRL 1–4 describe technology at levels of research, levels 5 and 6 describe technology on its way into product development and levels 7-9 describe technology on its way to the market.

There are no known commercial implementations of ABE. Therefore ABE is not described as TRL 7–9. Furthermore, TRL 5–6 do not describe the status of ABE in general, although some experiments are close in achieving this level.

Different ideas of how to design ABE policies and choose attributes are still discussed. Consensus has not been reached regarding these issues. Some environments, with expected needs indicating ABE as a suitable technology candidate, have been identified. Health care is one such environment where active research activities are making progress. Laboratory proof-of-concepts with restricted functionality have been made. At least within the health care sector efforts have been made to provide sector knowledge, needs and requirements as input to the ABE research. Technology components related to health care needs have to some extent been validated in a laboratory environment. Due to this, it seems reasonable to consider ABE to be somewhere in the interval TRL 1–4.

Whether ABE is appropriate to use in a certain environment is only partially indicated by the TRL level. However, several other aspects have to be considered before a mature method is achieved. One aspect to consider is ease of use versus complexity. ABE relies on complex mathematics. To make it usable in an operative environment, substantial efforts in user interfaces needs to be developed.

From the perspective of OBS, among the issues which will need further research and development are those related to key management, e.g. how to handle changes in groups of authorized users when people are added to groups or leaving and also of how to handle keys being lost.

4.4 A conceptual OBS-ABE architecture

The above identified benefits of using ABE in an OBS system, indicates the need to consider what a conceptual OBS solution based on ABE would look like. Therefore an architecture is proposed, which strives to utilize the potential of ABE in relation to the OBS vision.

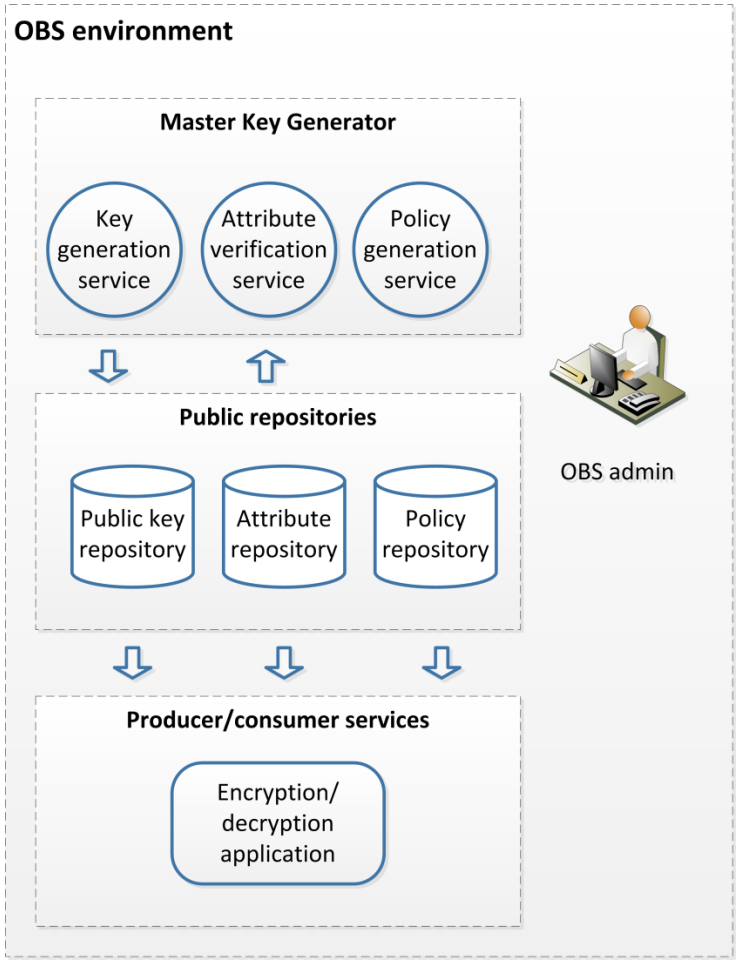

Figure 8 Proposed ABE-based OBS architecture.

The architecture, illustrated in Figure 8, is based on a producer – consumer perspective of handling information. Therefore it is convenient to describe the architecture as consisting of a number of services and repositories facilitating the communication activities of the producers and consumers requiring encryption and decryption. The architecture provides a high-level description, omitting details not needed to grasp the main characteristics of the communication activities and the entities involved.

The OBS environment constitutes four major pieces: the Master Key Generator (MKG), the repositories which store accessible information, a set of producers and consumers, and an OBS environment administrator (OBS admin).

The OBS admin manages and facilitates communication activities and with a specific responsibility of securely providing private keys for consumers. Access control, which is a main responsibility of the OBS administrator, regulates which information the consumer is allowed to access. For instance, it is the OBS administrator that decides which access structure a consumer should have in a KP-ABE policy private key, or which attributes to include in a CP-ABE policy private key. Although the issuing of providing private keys is here described as an offline activity, it is possible to implement key distribution as an online service. That decision is up to the security policy of the OBS environment.

A specific instantiation of the OBS environment may consist of several cooperating MKGs, a number of repositories, several sets of producers and consumers and multiple OBS administrators. This would most likely be typical in a coalition scenario.

There is a variety of ABE constructs that were presented in Chapter 3, but are not detailed in the architecture. ABE constructs such as HABE, multi-authority and hidden policies are also not explicitly described by the architecture. These constructs though, may still be an integral part of a more detailed design based on the architecture. To achieve an expected and required level of confidentiality and availability, for example within a coalition in operation, it seems likely and plausible that several such constructs may be needed.

The MKG in an OBS solution consists of three services:

- Key generation service
- Policy generation service
- Attribute verification service

The policy generation service is a piece of software designed to provide the MKG with an access structure for KP private keys. The OBS administrator provides input to the service. In the case of issuing CP private keys, the OBS administrator provides a set of attributes to the MKG which the attribute verification service verify match the controlled vocabulary of the attribute repository. The key generation service generates the private key of the consumers with input from either the attribute verification service or the policy generation service.

The MKG, as described in the ABE literature, is essential for any ABE system, since it generates public keys used for encryption under an access policy based on a set of attributes and the Master Secret Key used to generate private keys for the decryption of messages. Because of the key generation capability, the MKG is also the largest vulnerability within the system. This vulnerability can be dealt with through an administrative solution by putting the MKG offline and letting the OBS administrator be the only allowed point-of-contact. The feasibility of a solution like that depends of course on how the OBS environment is defined. An

enterprise solution, for instance the Armed Forces, is beneficial as it defines a relatively small group of individuals. A worldwide public solution puts more stress on the OBS administrator but it is still possible to maintain the MKG offline. If an online MKG is desired, the security requirements on the MKG will be tough.

One way of protecting the MKG is to make use of repositories, thus shielding the MKG from direct user interaction. The OBS environment suggests three repositories: Public key repository. Attribute repository, and Policy repository. The repositories are locally available resources which provide access to information that needs to be readily available, preferably without involving the MKG. The public key repository stores the MKG's public key(s). Any attribute-based or metadata system will need a controlled vocabulary – that is – a set of approved attributes with defined values. These attributes are stored in the attribute repository and are used by all participants in the OBS environment. Lastly, the policy repository contains predefined policies. In a CP-ABE context the producer defines the policy before encrypting a message. In an enterprise environment however, it would probably be preferable with a set of approved policies that follow the enterprise's information model or business policies. It also acts as support to the producer as it can be difficult to decide on a policy by oneself.

A producer is the user who creates an encrypted message and a consumer is a user who is a part of the group on which the policy in the ABE encrypted message applies.

Although only hinted in the architecture, the information flow is mostly from the MKG, via the repositories, to the producers – users. There are two possible exceptions to this flow. When the MKG is presented with a set of attributes and asked by the OBS administrator to generate a private CP-ABE key, the MKG may wish to validate the correctness of the attributes. Thus the MKG may communicate with the attribute repository. Normally though, it is only the OBS administrator who interacts with the MKG. Administrating key generation and distribution in an offline manner is a common solution with the purpose of maintaining the confidentiality of the secret key and the integrity of the system.

5 Discussion and conclusion

Availability is one of the founding pillars describing the properties of information security. It is however more common to focus on confidentiality as it is well understood and often has legal implications. Information systems that primarily preserve the confidentiality of the information is also more common, at least in the military environment. Maintaining confidentiality where it is needed is of course important, and the purpose of this report is not to contradict that in any way. However, in a business context, whether it is in the private sector or within government, not being able to discover or reach crucial information in a timely fashion may also harm an organization. Information technology has changed the way business is conducted and how information is shared. Information may need to be accessible over large distances, at any time of day and, sometimes, to temporary teams or users.

Traditional access control mechanism such as access control lists are very dependent on the identity of an authorized user, and that information is sought within the creator's controlled environment. Although the mechanisms have advanced, for instance, through Attribute-Based Access Control (ABAC), it is still a mechanism that guards the information before it is released.

In this report, the focus has been on how to use Attribute-Based Encryption (ABE) as a mechanism to achieve the vision of Object-Based Security (OBS). The most natural competitor for ABE is ABAC, as it too is an attribute and policy-based mechanism. The main difference between ABE and ABAC, as far as OBS is concerned, is where and when the actual access control decision is being enforced. ABAC, although being able to achieve a more fine-grained policy with near real-time attribute values, needs to make the decision before the file in question is released. ABE postpones that decision to the time where the file is going to be used – that is when it is decrypted. Thus, ABE encrypted files can be distributed outside the physical domain of the producer of the information. For information sharing purposes, this means that files can be stored securely on third party computers, i.e. in the cloud, or distributed offline via portable memory sticks. As long as only the receiver can provide a decryption key that corresponds with the policy or attributed associated with the file, no one else can access the file.

A relevant issue to discuss when it comes to attribute-based systems, whether it is ABE, ABAC or any similar technology, is the trust that can be associated with the attributes. How are attributes, i.e. the properties of a user or a file, securely associated with the user or file? In ABE, the attributes are associated with a key, either in the encryption key for KP-ABE or in the decryption key for CP-ABE. In ABAC, static attributes can be stored in a local repository but the "just-in-time" attribute values need to be collected at the time of the request.

In ABAC the flow of actions are consecutive and need to be performed before access to an object is granted. In ABE, the events are also consecutive but with a larger delay. The person who wants to decrypt an object can be in possession of it beforehand, regardless if the person has the correct key to decrypt the object.

The drawback of putting a policy in a key is the ability to keep the policy up to date. ABAC allows for a detailed policy and the use of timely attribute values. Current properties such as time or location can be used as input when sending attributes to an access control function, but unless the user can create keys on the fly such a current key is impractical. The attributes in an ABE policy should thus correspond to properties that will not change too quickly. As far as a detailed policy is concerned, ABAC seem to be a more versatile solution. Staying with the vision of OBS though, with an ABAC solution a connection to an ABAC server will be necessary.

Currently, ABAC is a more mature technology as it is better understood. Thus far, there has not yet been a commercial breakthrough for ABAC, but it is, for instance, implemented in XACML version 3. ABE is very promising as a technology to realise an OBS model, but at the moment however, ABE does not provide the versatility of ABAC systems. This does not mean that ABE is not useful, only that it is not mature enough to be the primary method of access control for an organization. However, it does provide rudimental access control in environments where the values of the attributes do not change on short notice and where regular online access control is not possible. Thus, there is a place for ABE within the OBS concept but further research is needed to provide a full solution for the OBS vision.

The need to provide access control in an offline environment is not exclusive to the Armed Forces. Portable computers and thumb drives are examples of equipment which transport information outside the physical security domain and which can be used without reconnecting logically with that domain. However, it is more likely for systems within the Armed Forces not to be able to connect to the Internet or, if Internet access exists, connect to systems within the Armed Forces infrastructure. Tactical systems are even further limited. In this situation, a system with offline security functionality is needed to provide access control and to protect information if a laptop or a thumb drive is stolen. By being an encryption technology, ABE both provides access control and preserves confidentiality. The latter holds if the files are stored encrypted and the security of the decryption key is sufficient. A regular access control system can only preserve confidentiality until a file is released.

5.1 Conclusions

ABE is an interesting candidate for OBS as it can provide both confidentiality and integrity of the information, as well as availability outside the security

domain of the information producer. If used properly, a consumer is challenged every time access to the information is requested, thus providing protection for data at rest.

However, ABE is not yet a fully mature technology. So far, it is mostly a research area with few or none large-scale experimental or commercial implementations. As such, it is an area well worth following but it is too early to provide requirements for ABE solutions when developing enterprise information systems. To its credit, ABE is the only technology thus far that fulfils OBS's vision of availability.

5.2 Future work

The key to any attribute-based system is to provide a policy which can be validated and attributes which can be used to provide an adequate policy. For OBS purposes, further research is needed to identify security relevant attributes and to develop more fine-grained policies.

References

Akinyele, J.A., Pagano, M.W., Green, M.D., Lehmann, C.U., Peterson, Z.N.J. & Rubin, A.D. (2011). Securing Electronic Medical Records Using Attribute-Based Encryption On Mobile Devices. In *SPSM'11, Proceedings of the 1st ACM workshop on Security and privacy in smartphones and mobile devices*. Chicago (IL), USA 17 October 2011, pp. 75–86. DOI: 10.1145/2046614.2046628

Attrapadung, N. & Imai, H. (2010). Dual-policy attribute based encryption – Simultaneous access control with ciphertext and key policies. *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, E93-A (1), pp. 116–125.

Bethencourt, J., Sahai, A. & Waters, B (2007). Ciphertext-Policy, Attribute-Based Encryption. In *SP'07, Proceedings of the IEEE Symposium on Security and Privacy*. Berkeley (CA), USA 20-23 May 2007, pp. 321–334.

Chase, M. (2007). Multi-Authority Attribute Based Encryption. In *TCC 2007, Proceedings of the 4th Theory of Cryptography Conference*. Amsterdam, Netherlands 21–24 February 2007, pp. 515–534.

Chase, M. & Chow, S.S.M. (2009). Improving Privacy and Security in Multi-Authority Attribute-Based Encryption. In *CCS'09, Proceedings of the 16th ACM Conference on Computer and Communications Security*. Chicago (IL), USA 9–13 November 2009, pp. 121–130. DOI: 10.1145/1653662.1653678

Chen, N. & Gerla, M. (2009). *Dynamic Attributes Design in Attribute Based Encryption*. http://nrlweb.cs.ucla.edu/publication/download/539/dynatt.pdf [2014-08-27]

European Defence Agency (n.d.). *Technology Readiness Levels*. http://www.eda.europa.eu/docs/default-source/procurement/ceds-fsp-trl-definitions.doc [2014-11-19]

Goyal, V., Pandey, O., Sahai, A. & Waters, B. (2006). Attribute-Based Encryption for Fine-Grained Access Control of Encrypted Data. In *CCS'06, Proceedings of the 13th ACM Conference on Computer and Communications Security*. Alexandria (VA), USA 30 October–3 November 2006, pp. 89–98.

Hohenberger, S. & Waters, B. (2013). Attribute-Based Encryption with Fast Decryption. In *PKC 2013, Proceedings of the 16th International Conference on Practice and Theory in Public-Key Cryptography*. Nara, Japan 26 February – 1 March 2013, pp. 162–179. DOI: 10.1007/978-3-642-36362-7_11

Hu, V.C., Ferraiolo, D., Kuhn, R., Schnitzer, A., Sandlin, K., Miller, R. & Scarfone, K. (2014). *Guide to Attribute Based Access Control (ABAC) Definition and Considerations* (SP 800-162). Gaithersburg: National Institute of Standards and Technology. DOI: dx.doi.org/10.6028/NIST.SP.800-162

Internet Corporation for Assigned Names and Numbers (ICANN) (2014). *Review of Trusted Community Representation in Root Zone DNSSEC Key Signing Ceremonies*. https://www.icann.org/en/system/files/files/tcr-dnssec-key-signing-21jan14-en.pdf [2014-10-06]

Kapadia, A., Tsang, P.P. & Smith, S.W. (2007). Attribute-Based Publishing with Hidden Credentials and Hidden Policies. In *NDSS'07, Proceedings of the 14th Annual Network and Distributed System Security Symposium*. San Diego (CA), USA, 28 Februari–2 March 2007, pp. 179–192.

Lewko, A. & Waters, B. (2001). Decentralizing Attribute-Based Encryption. In *EUROCRYPT 2011, Proceedings of the 30th Annual International Conference on the Theory and Applications of Cryptographic Techniques Advances in Cryptology*. Tallinn, Estonia 15–19 May 2001, pp. 568–588.

Li, J., Wang, Q., Wang, C. & Ren, K. (2009). Enhancing Attribute-Based Encryption with Attribute Hierarchy. In *CHINACOM 2009, Proceedings of the 4th International Conference on Communications and Networking in China*. Xian, China 26–28 August 2009, pp. 158–162.

Li, M., Yu, S., Zheng, Y., Ren, K. & Lou, W. (2013). Scalable and Secure Sharing of Personal Health Records in Cloud Computing using Attribute-based Encryption. *IEEE Transactions on Parallel and Distributed Systems*, 24(1), pp. 131–143.

Ming, Y., Fan, L., Jing-Li, H. & Zhao-Li, W. (2011). An Efficient Attribute based Encryption Scheme with Revocation for Outsourced Data Sharing Control. In *IMCCC, Proceedings of the First International Conference on Instrumentation, Measurement, Computer, Communication and Control*. Beijing, China 21–23 October 2011, pp. 516–520. DOI 10.1109/IMCCC.2011.134

Ostrovsky, R., Sahai, A. & Waters, B. (2007). Attribute-Based Encryption with Non-Monotonic Access Structures. In *CCS'07, Proceedings of the 14th ACM Conference on Computer and Communications Security*. Alexandria (VA), USA 29 October–2 November 2007, pp. 195–203.

Pang, L., Yang, J. & Jiang, Z. (2014). A Survey of Research Progress and Development Tendency of Attribute-Based Encryption. *The Scientific World Journal*, Volume 2014, Article ID 193426. DOI: 10.1155/2014/193426

Qiao, Z., Liang, S., Davis, S., & Jiang, H. (2014). Survey of Attribute Based Encryption. In *SNPD 2014, Proceedings of the 15th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing*. Las Vegas (NV), USA 30 June–2 July, pp. 1-6. DOI:10.1109/SNPD.2014.6888687

Sahai, A. & Waters, B. (2005). Fuzzy Identity-Based Encryption. In *EUROCRYPT'05, Proceedings of the 24th annual international conference on*

Theory and Applications of Cryptographic Techniques. Aarhus, Denmark 22–26 May 2005 pp. 457–473. DOI: 10.1007/11426639_27

Simmons, G.J. (ed.) (1992). *Contemporary Cryptology – The Science of Information Integrity*. Piscataway: IEEE Press.

Traynor, P., Butler, K., Enck, W. & McDaniel, P. (2008). Realizing Massive-Scale Conditional Access Systems Through Attribute-Based Cryptosystems. In *NDSS 2008, Proceedings of the 16th Annual Network & Distributed System Security Symposium*. San Diego (CA), USA 8–11 February 2008. http://www.isoc.org/isoc/conferences/ndss/08/proceedings.shtml [2014-11-18]

Wang, G., Liu, Q, Wub, J. & Guo, M. (2011). Hierarchical attribute-based encryption and scalable user revocation for sharing data in cloud servers. *Computers & Security*, 30 (5), pp. 320–331. DOI 10.1016/j.cose.2011.05.006

Wang, X., Zhang, J., Schooler, E.M. & Ion, M. (2014). Performance Evaluation of Attribute-Based Encryption: Toward Data Privacy in the IoT. In *ICC 2014, Proceedings of the IEEE International Conference on Communications*. Sidney, Australia 10–14 June 2014, pp. 725 – 730. DOI: 10.1109/ICC.2014.6883405

Waters, B. (2011). Ciphertext-Policy Attribute-Based Encryption – An Expressive, Efficient, and Provably Secure Realization. In *PKC 2011, Proceedings of the 14th International Conference on Practice and Theory in Public Key Cryptography*. Taormina, Italy 6–9 March 2011, pp. 53–70. DOI: 10.1007/978-3-642-19379-8_4

Yu, S. (2010). *Data Sharing on Untrusted Storage with Attribute-Based Encryption*. Diss. Worcester Polytechnic Institute, USA. http://www.wpi.edu/Pubs/ETD/Available/etd-071310-143310/unrestricted/Yu.pdf [2014-10-06]

Yu, S., Ren, K. & Lou, W. (2008). Attribute-Based Content Distribution with Hidden Policy. In *NPSec 2008, the 4th Workshop on Secure Network Protocols*. Orlando (FL), USA 19 October 2008, pp. 39–44. DOI: 10.1109/NPSEC.2008.4664879

Yu, S., Wang, C., Ren, K. & Lou, W. (2010). Attribute Based Data Sharing with Attribute Revocation. In *ASIA CCS'10, Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security*. Beijing, China 13–16 April 2010, pp. 261–270. DOI 10.1145/1755688.1755720

Zhou, Z. (2008). *Survey of Attribute Based Encryption*. http://www.docstoc.com/docs/97960936/Survey-of-Attribute-Based-Encryption [2014-08-27]

Zhu, S., Yang, X. & Wu, X. (2013). Secure Cloud File System with Attribute based Encryption. In *INCoS-2013, Proceedings of the 5th International*

Conference on Intelligent Networking and Collaborative Systems. Xi'an, China 9–11 September 2013, pp. 99–103.

Appendix A: Key-Policy ABE

In this appendix, a more detailed description of Key-Policy ABE (KP-ABE) is provided. The solution is enhanced with the use of a symmetric key for encryption of the plaintext (Wang et al. 2014) and shown in Figure 9.

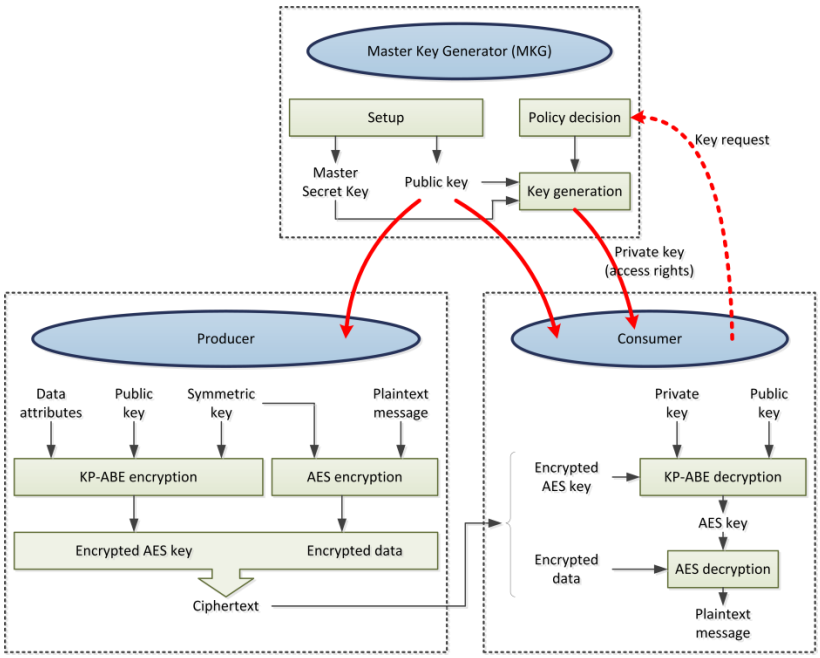

Figure 9 KP-ABE with an AES-encryption key (Wang et al. 2014).

The KP-ABE scheme consists of four algorithms (Goyal et al. 2006), here enhanced by Wang et al. (2014).

Setup. The Master Key Generator (MKG) initiates an ABE instance by generating a public key pair. The Master Secret Key is stored at the MKG while the public key is made available for producers and consumers within the MKG's security domain.

Encryption. A producer who wishes to publish a document encrypted by ABE choses a random symmetric key with which the plaintext message is encrypted. Then, the producer choses a set of data attributes and encrypts the symmetric key using the MKG's public key.

Key generation. The consumer of the encrypted message needs a decryption key to be able to decrypt the message. In order to obtain a decryption key, the

consumer requests one from the MKG. The MKG, in turn, generates a personal decryption key for the consumer, once the consumer has been authenticated. Within the decryption key, the MKG also includes an access structure which regulates which information the consumer is allowed to access. To be able to decrypt a message, the consumer also downloads the public key.

Decryption. With every component in place, the consumer can decrypt the ciphertext using the public key and the private key, thus retrieving the symmetric key. With the symmetric key accessible, the plaintext message is recovered.

Appendix B: Ciphertext-policy ABE

In this appendix, a more detailed description of Ciphertext-Policy ABE (CP-ABE) is provided. The solution is enhanced with the use of a symmetric key for encryption of the plaintext (Wang et al. 2014) and shown in Figure 10.

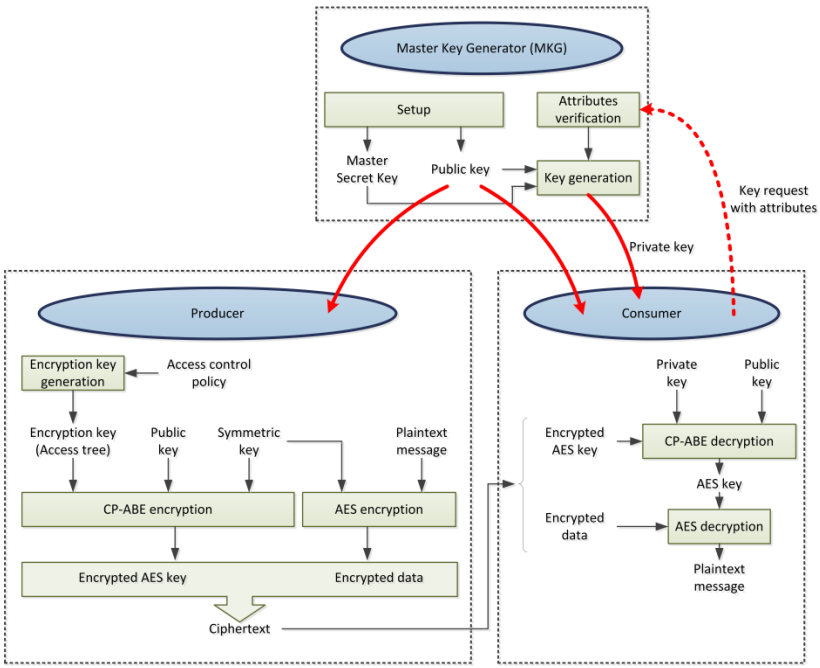

Figure 10 CP-ABE with an AES-encryption key (Wang et al. 2014).

The CP-ABE scheme consists of four algorithms (Bethencourt et al. 2007), here enhanced by Wang et al. (2014).

Setup. The Master Key Generator (MKG) initiates an ABE instance by generating a public key pair. The Master Secret Key is stored at the MKG while the public key is made available for producers and consumers within the MKG's security domain.

Encryption. A producer who wishes to encrypt a message using CP-ABE creates an access tree based on an access control policy. A symmetric key is generated and used to encrypt the plaintext message. The producer encrypts the symmetric key using the encryption key and the MKG's public key. The resulting ciphertext is then published.

Key generation. The consumer requests a key containing attributes which correspond to the consumer's current status. The MKG replies with a secret key (private key).

Decryption. The consumer is now able to decrypt the ciphertext, using the public key of the MKG, and the private key. The symmetric key is first decrypted, and then used to decrypt the actual message.

Object-Based Security (OBS) is a vision of information objects being able to carry with them a protective capability that preserves the properties confidentiality, integrity, and availability. Regular client – server solutions can fulfil the requirements for access control but requires that the consumer of the information is connected to the source of the information object. According to OBS, an access control function should provide service even if there is no connection to such a function. An information object which is transferred to a consumer, for instance via a thumb drive, provides the consumer with access to the actual information object but would, according to OBS, still need to be authorized before accessing the content.

In this report, a study of a technical candidate for the fulfilment of OBS is presented. The encryption method Attribute-Based Encryption (ABE) has been analysed through a set of questions which identify the OBS needs. In the report, a conceptual architecture is proposed which shows how ABE can be utilized to achieve an OBS solution.

FOI
Swedish Defence Research Agency
SE-164 90 Stockholm

Phone: +46 8 555 030 00
Fax: +46 8 555 031 00

www.foi.se