Secure Protocol of ABAC Certificates Revocation and Delegation

Alexey Rabin and Ehud Gudes

The Open University of Israel

Abstract. This paper deals with the maintenance of PKI certificates for Attribute Based Access Control (ABAC). We show, that the current standard has several problems in different revocation and delegation processes. This may lead to a security hole allowing usage of ABAC certificates, when it was revoked or transferred. As a solution we suggest architecture changes, that allow to perform revocation and transfer checks in such cases, based on extensions of the validation process of the ABAC certificates. We also discuss some privacy and performance challenges that are raised as a result of our proposal.

1 Introduction

The authorization process is one of the most important issues of the access control challenge. The classical approach of authorization is based on a concept of identification. Identification is a process that defines uniquely the subject that asks for permission, to the asset that provides it. Usually, the latter isn't defined specifically for a subject, but is bound to groups. The model of defining those groups and managing the mapping of subjects and permissions to them is usually referred to as RBAC - Role Based Access Control. RBAC is a De-Facto standard of authorization, and has many important advantages. However, there are some limitations in this approach, that make it hard to implement in certain scenarios. The most important limitation is its binary approach - the subject can only belong to, or not belong to a role. In scenarios where it is desired to calculate permissions using some logic, this approach is hard to implement. Another limitation of RBAC, is that a decision to add a subject to a role has to be driven by the actual permissions of this role, and not intuitively bound to the subject itself.

Another authorization approach, that is more flexible than RBAC, is ABAC (Attribute Based Access Control). This approach, introduced in [McCollum et al. 1990], suggests that the permission decisions will be taken by the asset based on the attributes' values of the subject. Unlike roles, attributes do not grant permissions directly, but try to describe the subject itself. The permission decision is based on two independent processes. The first is a description of a user by an Attribute Authority (AA) using several attributes. The second is the calculation that the asset manager performs, based on those attributes, which results with an access decision.

Figure 1 illustrates the basic authorization process of ABAC.

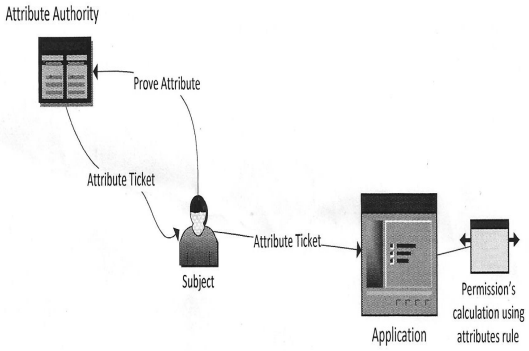

Fig. 1. Scheme of ABAC authorization

A subject asks the AA to prove some of its attributes. The AA returns a ticket that proves them. Now the subject can use this ticket to prove the attribute to the asset. When the asset gets the ticket's attributes, it uses the values as an input to an internal set of rules, which enable it to decide whether the asked permission should be allowed.

An important advantage of ABAC, which is critical to distributed environments, is that there can be a number of different AAs providing attributes. An asset can make decisions using reputation based calculations, such as introduced in [Xiong et al. 2004]. For example, an asset $Bank$ wants to calculate a credit rate of a subject $User$, who has asked for a loan. $User$ will be asked to provide attributes of $A_{MaritalStatus}$, of $A_{ChildrenNumber}$, and of $A_{AverageAccountSum}$ with value of daily average sum of money in the account belonging to the user. The $Bank$ can then assign importance of 5 to marital status, then to add to the credit rate, a 1 for each child but not to more than for 3 children, and to define thresholds for the average sum from -4 to 5. Final rank will be calculated by the sum of the values. Such attributes based calculations are an essential property of ABAC.

Modern ABAC environments also have the ability to provide the users with delegation and transfer of attributes. [Li et al. 2003] defines two delegation types, which are important to a full usage of decentralized environment. The first is the delegation of an attribute authority, i.e. trust of one entity on a judgment of another. An example of such a delegation is a situation where medical qualification of a doctor is proved to the patient by Ministry of Health of one country, based on his certification done in Ministry of Health of another. The second type of delegation, which extends the previous one, is attribute based delegation. This means that some AA will trust the judgment of another AA, if the latter has some attribute. An example of such a delegation is a situation where medical qualification of a doctor is proved to the patient by Ministry of Health, based on his certification done in any organization that has an attribute of Medical School. Unlike the previous case, the MOH doesn't necessarily trust the schools specifically, but trusts their certification.

An important enabler of the ABAC approach, that actually allows its decentralization, is the concept of Attribute Certificates (AC). The concept, firstly suggested in [McCollum et al. 1990] and standardized in [Housley et al. 1999, RFC2459], combines ABAC with PKI. In classical PKI, we use digital

signatures of CA (Certificate Authority) on a certificate, as an identity proof of the certificate's holder. In ABAC PKI, we use the digital signature of the AA on a certificate as a proof that the certificate holder has some attribute with a certain value. In addition to that, [Linn et al. 1999] suggested the use of anonymous certificates, i.e. ACs providing only the attributes themselves. This, in combination with a flexible protocol suggested in [Blaze et al. 1999], allows ABAC to be useful in scenarios where privacy is very important.
In the field of access control, permissions revocation is as important as permissions grant. ACs provide a simple ability to treat revocation process with tools of PKI, and those tools were adopted by the AC's standard [Farrel et al. 2010, RFC5755]. One of the differences of the latter standard from the previous ones is that it recommends not to use AC chains. Unfortunately, following this recommendation will not allow delegation process. On the other hand, as we will show in section 3, when AC chains are in use, revocation mechanisms of AC standards do not cover all cases in which the certificates shall be disabled. This includes the case of cascading revocations while inference property was used and in some cases of attribute transfer. Our paper will suggest ways to widen the revocation model of ACs, so that revocation will effectively work also in those cases.
One of the strengths of decentralized ABAC approach is its ability to provide good level of privacy. The AC contains only the information the asset needs to make authorization decisions, so minimal personal data is exposed to the asset. This data isn't sent to the attributes' provider when it is used by the asset, so the usage is not exposed to the data provider. When we suggest to make protocol changes in the AC standard it is important to us not to weaken this ABAC property. In sections 5 we explore the issue and suggest different solutions to this concern.

The main contribution of this paper is a solution proposal for the risks of inconsistency of cascade revocation and cascade delegation of ACs, which can be an enabler to secure usage of AC chains . In section 2 we present related work and discuss the current standards of AC. In section 3 we discuss the security inconsistencies of the revocation processes and suggest the infrastructure and protocol changes to treat them. In section 4 we present the needed algorithms' changes to support the model. In section 5 we present some challenges of our proposal and suggest approaches to solve them. We conclude in Section 6.

2 Related work

[Linn et al. 1999] provides the basis of modern design for Attribute Certificates. The main issue was the creation of certificates that prove some attribute and not only user identity, so the authorization decisions could be based on them. They also proposed a basic inheritance and revocation scheme, based on the usage of classical PKI inheritance and revocation. That work also suggested the use of anonymous certificates, i.e. a certificate that does not contain identity details, in

order to improve the privacy of the holder. This approach was explored further and created a framework such as U-prove ([Paquin et al. 2011]).
[Farrel et al. 2010, RFC5755] is the current standard for use of attribute certificates and it summarizes the work done in that field untill 2010. It also deals with revocation and delegation, and suggests the usage of classical PKI approach to them including the use of CRLs (Certificate Revocation Lists). Due to the fact that the administration of AC chains is complex, the standard recommends not to use them. In our work we show one consequence of that complexity - the current standard does not cover some possibilities of malicious use of ACs after revocation or transfer. We also suggest a solution to that security hole.
A totally different approach to revocation, introduced by [Rivest, 1998], suggested avoiding at all the usage of CRL. This is achieved by making the certificate life period short enough. The idea wasn't accepted in the classical PKI, but it was adapted for ACs in [Thompson et al. 2003]. In our work we don't rely on that technique, since there are practical scenarios when long living certificates are needed.
[Ye et al. 2006] propose a model for delegation, based on attributes, which is also important to our work. The model includes an attribute allowing to perform delegation of certain role and an attribute allowing to receive delegation of certain role. Though the work deals with the RBAC model, the approach allows decentralized delegation, based on the mentioned attributes, so the central manager, in advance, determines delegation rules and any participant can perform or receive delegation according to them. In ABAC this is a central idea.
[Crampton et al. 2008] defines several types of delegation, including strong transfer where the delegator loses all its rights after the transfer. We will show how strong transfer is incorporated in our model in the next section.
Earlier we mentioned that [Linn et al. 1999] proposed the usage of anonymous certificates. Those certificates have the benefit of privacy but create challenges in the revocation process, since the asset cannot connect a misbehavior to a certain user. The problem of creation of a protocol, that suggests correct balance between revocation efficiency and user privacy, is also a challenge in other fields of security research. One solution was suggested in [Lou et al. 2009], uses the idea of a Trusted Third Party (TTP), a proxy participant that masks the usage from different participants of the authorization negotiation process. Another approach is presented in [Win et al. 2012]. This work, that comes from the field of Digital Rights Management, proposes a scheme for revocation requests of assets, based on anonymous certificates. We will suggest to use both approaches as solutions to privacy concerns, later in our work.
Revocation check demands performance costs from the subject, and can even lead to Denial of Service attacks ([Hinarejos et al. 2010]). Improvements in efficiency of that check was a subject to wide research. In our work we will suggest to adopt ideas of CRL efficient structure of [Naor et al. 2000], of PREON algorithm of [Hinarejos et al. 2010] and of CREV-1 algorithm of [Yap, 2011] to make performance improvements of that process.
There is also a non-CRL revocation approach proposed by [Boneh et al. 2001],

which is based on key compromise solution of [Rivest, 1998]. This approach allows to minimize risks of DoS attacks and improve revocation performance. We plan to incorporate some of these ideas in future work.

3 Revocation Issues

Revocation is one of the most important issues in the PKI model. It deals with situations where the issuer decides that the certificate shall become not valid before it reaches its validity date. That can happen when new information is received about the subject or when his private key is lost or stolen.
In distributed environment the issuer has no way to know where the certificate could be used. That is why all PKI standards contain a way in which the asset can check certificate's revocation status. Usually, the check is based on CRL - a list of all revoked certificates, that is published by the issuer, and can be downloaded or checked on-line during login process. As we will show in this section, in case of ACs there are revocation situations where the check cannot be done. Therefore, there is a need to extend the classical CRL protocol to treat ABAC properly.

3.1 Inference Cascade Revocation

Inference means the ability of one AA to sign an AC for a subject, based on ACs it got from other AAs. The former AC is referred to as *result certificate*. The latter as a *reason certificate*. The value of a specific field of the result AC is calculated from values of the reason ACs. The process can be repeated. The chain of ACs can be described as a reversed tree with current AC as a root, vertexes as reason ACs, and arrows describing calculations.

In **Figure 2**, we see an example of such a tree. Att_p stands for some attribute. Att_1 and Att_2 are two attributes who's values' average is a value of Att_p. $Att1$ value is calculated as sum of Att_{11} and Att_{12} values. And Att_{12} value is calculated from Att_{121} value by some rule.

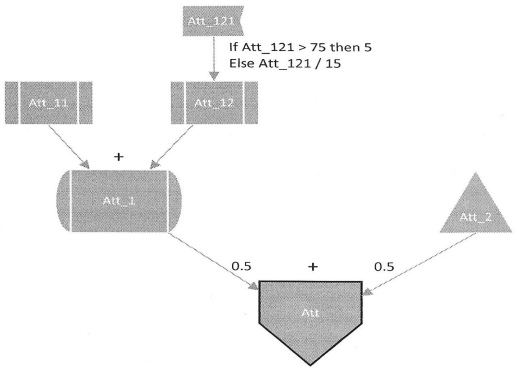

Fig. 2. Tree of attribute value calculation.

Revocation is a challenge when inference is used. In case one of the reason certificates is revoked and this certificate had proved some attribute that is

critical to the issuance of the result AC, the result certificate shall also be revoked. And if there is a path in an inference tree of critical reason certificates, we expect that revocation of AC in the deepest level will lead to revocation of all the path. Unfortunately, classical CRL protocols allow assets to check issuance chain but not inference chain. AC format, described in [Farrel et al. 2010, RFC5755] has no fields that show the inference connection. This is why the authority which issued the result certificate doesn't necessarily know about the revocation of the reason one, and therefore the derived certificates and all its descendants remain valid.

Problem Example : An attribute authority AA_{sch} gives certificates allowing work in schools to anyone who has B.Ed degree, proved by authority AA_{ed}, and has no legal violations, proved by authority AA_{police}. Some person can get the needed A_{ed} and A_{police} certificates, provide them to AA_{sch}, and get certificate A. According to current attribute certificate's standard, this certificate has no clear linkage to A_{ed} and A_{police}. It will be provided solely to assets (schools) as a permission to work. In case the holder is convicted in some criminal activity, his A_{police} certificate will be revoked. But since A has no link to its ancestor, it will remain valid.

One solution to the challenge, which follows the recommendation of [Farrel et al. 2010, RFC5755] not to use AC chains, is to put the revocation responsibility in such case on the shoulders of AA. The AA can monitor revocation of each reason certificates it ever used for its valid issuances, and when finding a revocation, to revoke the result certificates. But because of the overhead it puts on the AA and low effectiveness of the solution, we will suggest a different approach.

Our proposal to the issue is to add an extension to the current Attribute Certificate standard and create an infrastructure to support it. This extension is based on an idea that during the process of AC creation, an AA should define which certificates were critical to the AC construction, and consequently, their revocation shall lead to AC revocation. In our paper they will be referred to as *Mandatory Certificates - MC*. And since dependency decisions are transitive, all MCs of any Mandatory Certificate of a reason AC, shall also become MCs of the AC itself, and so on, recursively.

The certificate extension will contain a list of certificates' details. For each certificate we will store its ID and link to CRL. Those fields allow an asset to check the revocation of each Mandatory Certificate of the AC. Since the process is done against each CRL server, the Certificate ID will be unique. Consequently, the scenario of a user using a certificate that is based on the revoked one, will be prevented. We will not save details of the certificate itself but only the required meta-data, in order to keep better level of privacy.

3.2 Transfer Delegation Revocation

Delegation is one of the most important properties of ABAC, since it allows decentralization of access control. [Linn et al. 1999] propose a PKI implementation to the delegation process. In their suggestion, a delegator issues a certificate to a delegatee. This certificate can contain partial or full attributes the delegator

has. When the delagatee uses this certificate to get authorization from an asset, he should preserve not only his certificate, but also all the delegation chain starting from some trusted AA. The PKI delegation is combined in [Farrel et al. 2010, RFC5755] with the idea of delegation attributes of [Ye et al. 2006]. When the delegation process is allowed, the AA adds to the AC an attribute that indicates that. It can also add a set of rules for delegation (for instance - maximal number of allowed delegations or delegation type). When delegation happens, the delegator shall fulfill the rules defined in his certificate, and add the same or stricter delegation rules to the delegatee certificate. The asset that checks a delegated attribute, also checks the legality of all the delegation path - signature of each certificate by its parent, its revocation status and its legality against the rules defined by the parent.

Transfer delegation. [Crampton et al. 2008] defines different types of delegation, and specifically *transfer delegation*. In transfer delegation a delegator loses the transfered attribute when the delegatee gets it. In centrally managed environment it can be done easily by indication of central authorization authority. But in PKI standards the transfer operation is not defined. AC delegation, that is implemented via processes of certificate issuance and revocation actually allows the delegator to continue using his certificate after transfer or to transfer it to more than one delegatee. Another problem can happen when the AA wishes to demand strong transfer delegation. According to [Crampton et al. 2008] when strong transfer occurs, all attributes that were given because of the transfered attribute, will be revoked. The problem is similar to the Inference cascading revocation issue: Since the asset doesn't know that the reason certificate was transferred, it accepts the result AC that should have been revoked.

The easiest way to deal with transfer delegation problems is not to use AC Transfer delegation. In real life this type of delegation is not used widely. However, we believe that transfer extension adds important advantages in attributes' usage versatility and flexibility.

In order to solve the transfer delegation revocation issues we suggest to make two extensions to the protocol.

The first one demands creation of a suitable infrastructure. We propose here the concept of Certificate Transfer List(CTL), which is similar to Certificate Revocation List of PKI. In our proposal each Attribute Authority that issues a transferable certificate adds into it the field of CTL location. The owner of an attribute certificate gets the permission (by its attribute certificate) to add the transfer fact to the list or delete it when the transfer is stopped. The transfer list also contains the field of ID and public key of a new certificate.

When the asset gets a certificate that can be transferred, it first checks whether a transfer was already done. If yes, it checks in the CTL whether the transfer was actually done, and whether it was done to this subject. If a chain of transfers was performed, all the chain should be written to the CTL and all CTL locations of the chain must be the same. In that way, at any moment there can be only one valid certificate with the given attribute. The transferrer can use the attribute

only before he writes it to the CTL, and the transferee can only use it after. Issuance of more than one certificate can't be done, as the transferee certificate virtual identity is also part of the Transfer List.
In order to allow Strong Transfer and to allow transfer treatment in cascade inference certificates, we propose to widen the Inference extension discussed earlier. Unlike the case of regular certificates, transfered AC shall be treated as chains. That means that mandatory certificate list shall contain, in addition to regular certificates' links discussed earlier, also members of type transfer certificate. This kind of member is built of the fields of common CTL, pairs of ID and CRL, and a binary field denoting whether it is Strong Transfer. The asset that checks a transfer certificate shall scan the CTL and check that all the chain exists, all certificates are valid, and that the last ID wasn't transfered.

Figure 3 illustrates an example of a table that can be added to the AC according to our proposal. The treated scenario is of Reputation AA that provides certificates of financial risk. In this specific case, the central parameters for risk calculation were person's clearance status (proved by AC from Police Department AA in the first row), his education level (proved by AC from Ministry of Education AA in the second row, that was given based on his degree proved by AC from the Open University AA in the third row), real estate he owns (proved by strong transferable AC from Land Authority AA in the fourth row), the car he owns (proved by non-strong transferable AC from Ministry of Transport AA that was issued to someone else and transfered to him in the fifth row), and details of the bank account he uses (proved by strong transferable AC from Free Bank AA, that was issued to one person, transfered to another and then transfered to our subject). As we suggested, the rows have only IDs and links to relevant lists, so no actual value of any attribute is exposed.

ID	CRL	CTL	Transfer List	Strong
4568625	https://police.gov.il/clearance/crl			
7853264	https://edu.gov.us/highschool/degrees/crl			
8924684	https://openu.ac.il/staff/crl			
9256246	https://land.gov.il/owners/crl	https://land.gov.il/owners/ctl	-	TRUE
85236472	https://transport.gov.il/owners/crl	https://transport.gov.il/owners/ctl	9374903 → 85236472	FALSE
85426642	https://FreeBank.com/accounts/crl	https://FreeBank.com/accounts/ctl	8248216 → 546582 → 85426642	TRUE

Fig. 3. Table of Mandatory Certificates.

The proposed changes are incorporated in the algorithms which are described and explained in the next section.

4 The New Algorithms

Our proposal leads to changes of four algorithms in the AC model.
Certificate creation should be changed to allow addition of mandatory certificates

table to the AC. **Certificate delegation** process should be changed to include registration of transfer fact in the Transfer List. The **delegation revocation** process should be changed to include transfer status change in the CTL. **AC validity check** process should be changed to include checks of correctness of all MCs. **Regular certificate revocation** process remains the same as in classical AC protocol and therefore won't be added here.
We assume here that all CTLs and CRLs are accessible. The other case shall be treated according to asset policy and can vary in different cases.

Next we present the algorithms.

4.1 Certificate creation

This algorithm is performed by the Attribute Authority. It doesn't deal with processes of attributes calculation and specific fields values (which is unique to each AA and each type of certificate), but only with creation of fields needed for consistency checks. The algorithm builds the MC table of a new certificate.Instruction 3.1 sets the validity of AC to be minimal of all its MCs. Instruction 3.2 adds the MC table of each MC to the table of AC (so recursively all inference chains are added).The table is built of MCs and of their MC tables' members. For transferable certificates the CTL is added to the MC. Sections 3.4 - 3.5 treat transfer : Signs whether the transfer is strong and add the transfer chain to the certificate.

```
1. Create all needed attributes
2. Define Mandatory Certificates
3. For each Certificate in  Mandatory Certificates :
3.1    ValidUntil = min (ValidUntil, Certificate.ValidUntil)
3.2    For each Line in Certificate.MandatoryCertificatesTable
                 Add Line to  MandatoryCertificatesTable
3.3    Add [ Certificate.ID, Certificate.CRL]
to MandatoryCertificateTableLine
3.4    If Certificate isTransferable and TransferType = Strong
3.4.1       Add Certificate.CTL to CertificateLine.CTL
3.4.2       Add all certificates IDs in transfer chain
to CertificateLine.TransferChainList
3.4.3        CertificateLine.isStrongTransfer = TRUE
3.5   Else If Certificate is Transferred
3.5.1        Add Certificate.CTL to CertificateLine.CTL
3.5.2        Add all certificates IDs in transfer chain
                 to CertificateLine.TransferChainList
3.5.3        CertificateLine.isStrongTransfer = FALSE
3.6  Add CertificateLine to MandatoryCertificatesTable
```

Example : We will follow the example illustrated in Figure 3. Instruction 3.2 is relevant to AC given by Ministry of Education. Since this AC has MC table

of his own, with AC given by Open University there, this Open University line will be copied to current MC table. Instruction 3.3 adds CRL locations of all ACs. Instruction 3.4 is relevant to Land AC and Free Bank AC. For both the CTL and the sign of strong transfer are written via instruction 3.4.1. For Free Bank, the attribute is itself transfered, and therefore instruction 3.4.2 copies the IDs of its Attribute Transfer chain (8248216 that issued 546582 that issued the current certificate 85426642) to the Transfer List field. Instruction 3.5 is relevant to Transport certificate, which is transfered, though it isn't strong. Instruction 3.5.1 saves its CTL, and 3.5.2 its PKI chain (9374903 that issued current 85236472). See 4th line in the figure.

4.2 Certificate Delegation

This algorithm is performed by a delegator in order to make delegation. The delegator creates and signs the delegatee certificate, and, in case of transfer delegation, adds the fact to the CTL. Re-delegation is treated in the same way.

```
1. Create delegateeCertificate
2. Add delegatorCertificate to delegateeCertificate.SignersChain
3. If delegation type = Transfer
3.1   If delegator already transfered certificate Return Error.
3.2   Else
3.2.1       delegateeCertificate.CTL = delegatorCertificate.CTL
3.2.2       Add [delegatorCertificate.ID, deleateeCertificate.ID,
               delegateeCertificate.signature] to
               TransferList  -> delegatorTable
4. Sign delegateeCertificate
```

Example : We will follow the example of FreeBank AC illustrated in Figure 3. Free bank gave the AC of the account owning to ID 8248216. That owner transfered it to 546582, and the latter to 85426642. Each delegator constructed and signed the delegatee AC via instructions 1,2. Since instruction 3 is relevant, instruction 3.2.1 adds link to CTL (https://FreeBank.com/accounts/ctl) to ACs of delegatees. After the two transfers, the FreeBank CTL will have the section for 8248216 transfers. Instruction 3.2.2 will add two lines : $8248216 -> 546582$ and $546582 -> 85426642$. see line 6 in the figure.

4.3 Delegation Revocation

This algorithm is performed by a delegator that wants to stop the delegation. The revocation is done by deleting all transfers from the delegatee and later from the CTL. This way each certificate transfered later will be considered invalid by the CTL check.

```
1. Add delegatee.CertificateID to delegatee.CRL
2. If delegationType = Transfer
2.1       Delete all lines in TransferList
from delegatee and further down
```

Example : We will follow the above example of FreeBank AC illustrated in Figure 3. If 8248216 decides to stop the transfer, both CTL lines will be deleted.

4.4 Certificate Validation during Access Check

This algorithm is part of access check algorithm. It is performed by the asset. After it is finished correctly, the asset can use the values of the attributes signed in the certificate, to calculate the subject's permissions. The algorithm checks the general certificate validity, and for certificates that were transfered to the subject, it checks whether the transfer is still in the CTL, and for strong transferable certificates, it checks in the CTL that they were not transfered elsewhere.

```
1. Make full X.509 validity checks. If failed return FALSE.
2. For each Certificate in Signers chain from current to root
2.1    If Parent.mode = Transfer
2.1.1     If Parent.Certificate.ID not in (Parent.Certificate.CTL)
 return FALSE
2.2.   For each CertificateID in Mandatory Certificate Table
2.2.1        If CertificateID in list CRLLocation return FALSE
2.2.2        If CTL exist
2.2.2.1         If the transfer order in CTL is NOT the same as in
           TransferChainList return FALSE
2.2.2.2         If isStrongTransfer and
CertificateID in CTL and not last
return FALSE
3. Return TRUE
```

Example : We will follow the example illustrated in Figure 3, with different revocation scenarios. Let's assume that reason AC of our Reputation AC, direct or indirect, is revoked. Since all of them are in MC table constructed by algorithm 4.1, instruction 2.2 will check their CRL, and 2.2.1 return revocation error.
Let's assume that the subject transfered his land ownership to somebody else. Algorithm 4.2 will create line in Land Authority CTL for this transfer. If he tries to use our Reputation AC, instruction 2.2.2 will check the CTL of Land Authority, and instruction 2.2.2.2 will find out that current AC is a delegator, so the algorithm will return revocation error.
Let's assume that the owner of Free Bank account decides to stop its transfer. He will change the CTL, as demonstrated in section 4.3. If this happened before Reputation AA started his work, instruction 2.1.1 will return validity error and the AA won't build the certificate. However, if reputation AC already exists, the asset validity check will find the revocation via instruction 2.2.2.1.

5 Challenges and Solutions

In this section we show different challenges that are raised as a result of our proposal and suggest solutions to them.

5.1 Privacy and Confidentiality Challenge

As we've mentioned before, ABAC authorization process involves three actors : the subject, the asset and the AA. Any actor is interested in minimal disclosure of its personal data to others. The subject is interested in *Privacy*. Therefore, it would like to prevent the asset from getting any of its attributes that are not explicitly needed to authorization decisions, and to prevent the AA from discovering the usage of the attributes it issued. The AA is interested in *Confidentiality*. Therefore it would like to prevent the asset from discovering the values' calculation algorithm, and specifically, which attributes were used for the calculation. The asset is interested in *Confidentiality and Privacy*. Therefore it would like to prevent the AA from discovering which attributes were used and why.

Metadata Challenge Our suggestions contain additions to the AC format that create a Privacy challenge. Unlike in a standard AC, our AC contains fields that point to CRLs and CTLs of MCs, and this data is available to the asset. As a result, there are a few scenarios that can lead to abuse, even if the AC is anonymous:
First scenario is of a malicious asset, which can try to find more information about a subject than required. A key to such an abuse is the fact that the link to the CRL contains meaningful meta-data : Who certified the attribute. This meta-data allows to discover authorities the AA relies on. In certain cases it is easy to guess which attribute it can be, and respectively to see which attributes the subject has. Moreover, a comparison of different ACs can help the asset to guess how the inference attributes influence the result that the AA produces, and to compromise the AA confidentiality.
Another scenario is of a malicious AA. When the AA gets a request for a CRL check from the asset, it finds out what is the subject's usage for the certificate (whether direct or by inference), violating subject's privacy. In addition, the AA that is a part of an inference chain, and gets a CRL check from a latter AA in the process of construction of the final AC, can easily use it to understand the trust algorithm of the asset and of the different AAs.

Revocation Request Challenge [Win et al. 2012] exposed a built-in collision between the demand to anonymity and the ability to make an efficient revocation. In our work we assumed that the revocation situations were initiated by the AA when it got some external information about a user. However, there are common scenarios of revocation, which involve a report of some assets to the AA, that certain subject has misused the credentials he got. The same is true in case of delegation. Unfortunately, such a report can enable the AA or the delegator to understand for which purposes the AC was used (or tried to be used). This can violate the subject's privacy and the asset confidentiality.

5.2 Performance Challenge

Personal certificates have only little impact on performance problems, since they are not used very often, and usually only during the login process. Attribute certificates are different - they can be used for any authorization demand. Therefore, the performance of the certificate check process is important. Revocation checks are time consuming, especially because of network problems. Our proposal demands increase in time consumption in three ways.
The first is by the fact that MC table is added and should be treated in each check. This increase is tiny, since it is treated entirely by and in the SP.
The second is in transfer revocation process, where in case of re-delegation the full chain of transfer shall be deleted. This increase is also very small, since it is treated entirely in the CTL, happens only once and adds only one network connection.
The third and the most significant impact is caused by the CRL and CTL check. Each row of MC table invokes such a check and demands separate network connection. Although those connections can be treated parallely, we still add some impact on the revocation process, and wait for the last answer.
In extreme cases, the check process can be even used as a mechanism for a denial of service attacks. All CRL solutions are known for being sensitive to such attacks, especially in case of a single publish server ([Adams et al. 1998]). In the AC case, as long as there are more validations than with regular certificates, as they are more complicated (for CTL), and are done more often - the danger of DoS is larger.

5.3 Challenges solutions

In order to address the above challenges we suggest two different approaches. One is based on Trusted Third Party and is more centralized, and another on Semi-Trusted Mediator and more connected to certificates solution. We will show how they can solve the above issues and what are the advantages of each one. However, the full comparison between them, and a detailed formalization of their usage is beyond the scope of this work.

Trusted Third Party TTP is a concept suggested by [Lou et al. 2009]. It means an external revocation proxy. Its idea is to prevent the case where any single actor knows enough information to abuse it for additional knowledge extraction. In our case, during AC creation, the AA writes the real MC table to some proxy server, and put in the AC links to that server. When an asset validates the AC, it will turn to the TTP, which will look for valid CRLs and CTLs and return a result.
The TTP is also a good platform for misbehavior reports treatment. When an asset suspects a problem with the AC, it can report it to the TTP. The TTP shall save statistics of such reports, and report to the AA that created a problematic AC, when their severity, number or any other metric reaches the predefined threshold. The AA will get the suspicions but not the assets that raised them.

In order to solve the performance issues, the TTP can act as a cache proxy. Unlike the regular proxy, the cache proxy contains not only links but also the data itself. In our case the TTP will periodically sample each MC CRL and CTL and save the general AC status. During the validation process the asset will have to make only single and simple status checks against the TTP and not all the checks against all MCs' AAs.
Cache proxy has the risk of revocation time increase. Therefore it is important to reduce it. We suggest a combination of two techniques. The first is a usage of CREV-1 algorithm described in [Yap, 2011] to notify the TTPs with changes. The other is a stateless subscription TTPs to CRL and CTL changes, as described in [Naor et al. 2000]. If TTPs get revocation notifications, in both ways, from the AAs and other TTPs, the risk becomes lower.

Semi-Trusted Mediator The SEM concept in classical PKI, suggested by [Boneh et al. 2001] is very different from the regular CRL. It suggests to prevent the need in revocation publisher via the usage of dual-stage PKI scheme - half of the keys is given to the user and half to an on-line semi-trusted server. Both halfs are needed to encrypt or decrypt a message. When some certificate is revoked all such servers are instructed not to cooperate with its holder anymore.
An adaptation to the AC case will demand an AC holder to get validity tokens for the AC itself and for each MC. All those tokens will be given to the asset during authorization process, as a validity proof. To allow that, the AA will create another pair of keys for each AC and give them to SEM.
In our adaptation we would use the concept of accumulator. It was firstly introduced by [Benaloh et al. 1993], and adapted by [Camenisch et al. 2002] and [Camenisch et al. 2009], to an efficient revocation technique for anonymous credentials. The model is based on Zero Knowledge proof of a validity token (witness). In our case each SEM will provide accumulator service, and the MCs of an AC will be treated similarly to credentials treatment in the original model. An important advantage of the concept is that most of the validity work is done by the client. Consequently, performance issues of an asset are reduced significantly and denial of service attacks are hard to implement. Another advantage is the fact that there is no need to use the publish infrastructure. Note also that though the performance concern is in general reduced, the SEM approach still requires more computational overhead because of the cryptographic operations during validity check.
The concept also helps to reduce privacy concerns, as there is no direct connection between the assets and the AAs. Therefore scenarios of AAs that understand the usage of ACs are prevented. In order to find a complete solution to the meta-data concerns, we suggest a naming convention for semi-trusted servers that doesn't involve AA information and usage of the same SEM for different AAs.
The issue of misbehavior report is not solved natively in this approach. In order to treat it we suggest to send reports to the SEM server. We also propose that during the creation of an AC, the AA shall instruct the SEM on which misbehavior thresholds to stop cooperate with an AC.

6 Summary and Future Work

ABAC certificates enable decentralized and flexible mechanism for treatment of the authorization process. Currently their revocation and transfer processes can lead to inconsistency, and therefore should be limited. Our proposal allows to solve this inconsistency and provides a flexible protocol for revocation and delegation. On the other hand, the proposal raises privacy and performance concerns, and we presented some ideas for their treatment. It is also a base for future work that can extend the current model to become a new full secure model for ABAC certificates.

A basic further work on the suggested model is its extension to more complicated cases of dependencies between certificates. Our current model treats a basic case where any ancestor revocation shall lead to a descendant revocation. However, it is not always the case, due to the fact that ABAC allows the AA to make decisions based on a group of provided certificates. It is clear that a correct treatment of complicated dependencies will involve creation of an assertion language that shall allow the AA to create description of checks to revocation. This language will have all classical logical operators, such as OR, AND, One Of, etc. In addition, further work shall explore an issue of a treatment that will reduce the ability of a conspiracy between different actors in order to find more information. Our model prevents an ability of a single actor to abuse the data it gets, but it can and should be widened to enhance confidentiality and privacy of all the actors when different participants collude to get extra information.
In general, we believe that our suggestions and their future extensions can make AC usage more flexible and secure, and therefore allow ABAC approach to be accepted in scenarios in which it is limited today.

References

[McCollum et al. 1990] McCollum, C. J., Messing, J. R., Notargiacomo, L. . Beyond the pale of MAC and DAC-defining new forms of access control.Research in Security and Privacy, 1990. Proceedings., 1990 IEEE Computer Society Symposium on (pp. 190-200). IEEE. (1990)

[Adams et al. 1998] Adams, C., Zuccherato, R. (1998). A general, flexible approach to certificate revocation. Entrust Technologies White Paper.

[Blaze et al. 1999] Blaze, M., Feigenbaum, J., Keromytis, A.D. : KeyNote: Trust management for public-key infrastructures. Security Protocols (1999). Springer Berlin - Heidelberg, 1999.

[Rivest, 1998] Rivest, R. L. Can we eliminate certificate revocation lists?. In International Conference on Financial Cryptography (pp. 178-183). Springer Berlin -Heidelberg, (1998).

[Housley et al. 1999, RFC2459] Housley, R., Ford, W., Polk, W., Solo, D. : RFC 2459 : Internet X.509 Public Key Infrastructure Certificate and CRL Profile (1999).

[Boneh et al. 2001] Boneh, D., Ding, X., Tsudik, G., Wong, C. M. A Method for Fast Revocation of Public Key Certificates and Security Capabilities. In USENIX Security Symposium (pp. 22-22). (2001, August)

[Li et al. 2003] Li, N. , Grosof, B.N. , Feigenbaum, J. : Delegation logic: A logicbased approach to distributed authorization. ACM Transactions on Information and System Security (TISSEC) 6.1 : 128-171. (2003).

[Linn et al. 1999] Linn, J., Nystrom, M. : Attribute certification: an enabling technology for delegation and role-based controls in distributed environments. Proceedings of the fourth ACM workshop on Role-based access control (1999).

[Xiong et al. 2004] Xiong, L., Liu, L. : Peertrust: Supporting reputation-based trust for peer-to-peer electronic communities. IEEE transactions on Knowledge and Data Engineering 16.7 (2004).

[Ye et al. 2006] Ye, C. , Wu, Z. , Fu, Y : An attribute-based delegation model and its extension. Journal of Research and Practice in Information Technology 38.1 (2006): 3-18 (2006).

[Naor et al. 2000] Naor, M., Nissim, K. . Certificate revocation and certificate update. IEEE Journal on selected areas in communications, 18(4), 561-570. (2000).

[Farrel et al. 2010, RFC5755] Farrell, S. , Housley, R., Turner. S. : RFC 5755 : An Internet Attribute Certificate Profile for Authorization. IETF (2010).

[Thompson et al. 2003] Thompson, M.R. , Essiari, A., Mudumbai, S. : Certificate-based Authorization Policy in a PKI Environment. ACM Transactions on Information and System Security (TISSEC), 6.4: 566-588 (2003).

[Lou et al. 2009] Lou, W., Ren, K. : 2009. Security, privacy, and accountability in wireless access networks. IEEE Wireless Communications, 16.4: 80-87 (2009).

[Win et al. 2012] Win, L.L., Thomas, T., Emmanuel, S. : Privacy enabled digital rights management without trusted third party assumption. IEEE Transactions on Multimedia , 14.3: 546-554 (2012).

[Yap, 2011] Yap., R.H. : Trusted principal-hosted certificate revocation. IFIP International Conference on Trust Management, pp.173-189. Springer Berlin - Heidelberg, 2011.

[Hinarejos et al. 2010] Hinarejos, M.F., Munoz, J.L., Forne, J., Esparza, O. : PREON: An efficient cascade revocation mechanism for delegation paths. Computers and Security, 29.6: pp.697-711 (2010).

[Crampton et al. 2008] Crampton, J., Khambhammettu, H. : Delegation in role-based access control. International Journal of Information Security 7.2, 123-136 (2008).

[Paquin et al. 2011] Paquin, C., Zaverucha, G. . U-prove cryptographic specification v1. 1. Technical Report, Microsoft Corporation. (2011)

[Benaloh et al. 1993] Benaloh, J., De Mare, M. One-way accumulators: A decentralized alternative to digital signatures. In Workshop on the Theory and Application of of Cryptographic Techniques (pp. 274-285). Springer Berlin Heidelberg. (1993)

[Camenisch et al. 2002] Camenisch, J., Lysyanskaya, A. . Dynamic accumulators and application to efficient revocation of anonymous credentials. In Annual International Cryptology Conference (pp. 61-76). Springer Berlin Heidelberg. (2002)

[Camenisch et al. 2009] Camenisch, J., Kohlweiss, M., Soriente, C. . An accumulator based on bilinear maps and efficient revocation for anonymous credentials. In International Workshop on Public Key Cryptography (pp. 481-500). Springer Berlin Heidelberg. (2009)