FFI-rapport 2009/02293

Cross-domain access control in a military SOA

Nils Agne Nordbotten

Norwegian Defence Research Establishment (FFI)

22.12.2009

FFI-rapport 2009/02293

1086

P: ISBN 978-82-464-1762-2
E: ISBN 978-82-464-1763-9

Keywords

Sikkerhet

Aksesskontroll

Identitetshåndtering

Tilgjengelighet

SOA

Approved by

Anders Eggen	Project Manager
Eli Winjum	Director of Research
Vidar S. Andersen	Director

Summary

A service-oriented architecture provides the advantage of facilitating integration and information exchange between different administrative domains (e.g., between the nations within a coalition). In order to support secure and efficient information exchange and service invocation between such domains, cross-domain access control is required. A collection of domains with such a capability is often referred to as a federated system and several civilian solutions and standards have been established for this purpose.

This report considers the most common architectural approaches (i.e., communication patterns) used for providing cross-domain access control in a SOA. It is shown that these existing solutions designed for use on the Internet, may not be suitable for use in military systems due to the differing connectivity characteristics and availability requirements. This is particularly the case if horizontal cross-domain interaction and information exchange is to be provided at the tactical level. In order to mitigate this, we propose an alternative approach being less dependent on strong connectivity. Independent of the chosen solution though, there is a tradeoff between the availability of services and making sure that access control decisions are based on correct (i.e., up to date) information about the requester.

Because existing solutions may not provide the sufficient assurance for use in military applications, this report also discuss how higher assurance requirements can be met while still to a large extent making use of existing standards and implementations.

Sammendrag

En tjenesteorientert arkitektur legger til rette for integrasjon og informasjonsutveksling på tvers av administrative domener (for eksempel mellom de ulike nasjonene i en koalisjon). For å kunne understøtte dette på en effektiv og sikker måte er man avhengig av en felles løsning for aksesskontroll på tvers av domener.

Det finnes flere sivile standarder for dette formålet. Denne rapporten ser på de vanligste kommunikasjonsmønstrene brukt av løsninger for aksesskontroll (og tilhørende identitetshåndtering) i SOA. Disse er i utgangspunktet tiltenkt bruk på Internet og andre nettverk med høy konnektivitet, og er ikke nødvendigvis egnet for bruk i militære systemer med varierende konnektivitet og høye krav til tjenestetilgjengelighet. Dette er spesielt tilfelle dersom en skal støtte horisontal integrasjon og informasjonsutveksling mellom domener på taktisk nivå. En alternativ løsning med lavere krav til konnektivitet foreslås derfor i rapporten. Uavhengig av løsning vil det imidlertid måtte bli et kompromiss mellom tjenestetilgjengelighet på den ene siden og det å forsikre seg om at aksesskontroll utføres basert på oppdatert informasjon på den andre siden.

Da eksisterende implementasjoner av de relevante standardene er relativt store og komplekse vil de ikke nødvendigvis kunne tilby den tilliten som kreves av sikkerhetsmekanismer i militære systemer. Rapporten diskuterer derfor også hvordan høyere tillit kan oppnås selv om man i stor grad baserer seg på eksisterende standarder og implementasjoner.

Contents

List of abbreviations

IPSec	Internet Protocol Security
MILS	Multiple Independent Levels of Security
SAML	Security Assertion Markup Language
SOA	Service-Oriented Architecture
XACML	eXtensible Access Control Markup Language
XML	eXtensible Markup Language

1 Introduction

A service-oriented architecture (SOA) provides the advantage of facilitating integration and information exchange between systems in different administrative domains (e.g., within a coalition). However, such cross-domain interaction can not be allowed unless the appropriate access controls can still be enforced. A cross-domain solution for access control is therefore required. Several civilian standards that can be used for this purpose exist. In order to facilitate interoperability and limit costs, it is an advantage if these standards can be used in military systems as well.

The eXtensible Access Control Markup Language (XACML) [1] provides an XML language for specifying access control polices. Using XACML, access control policies and decisions may be based on the attributes of the subject (i.e., the user, device, or other entity requesting access), the resource, the environment and/or the action to be performed. Such a model is sometimes referred to as attribute based access control [2], and provides a high degree of flexibility when defining access control policies. The attributes of a subject may for instance include its roles, clearance level, age, and/or unique identifier. Likewise, environment attributes may for instance include the time of day, the current threat level, or the current load.

Because attributes can be used to specify both roles and identities, such an access control model encompasses both identity and role based access control.[1] Furthermore, such a model may also be used to enforce mandatory access control according to the Bell-La Padula model [3]. Hence, without loss of generality, we will assume that some variation of attribute based access control is utilized.

In order for attribute based access control to be performed in cross-domain scenarios, the attributes of the subject must be communicated from the administrative domain of the subject to the domain of the resource. The Security Assertion Markup Language (SAML) [4] defines an attribute statement by which the attributes of a subject can be expressed within a SAML assertion, and communicated in a secure manner. Such an attribute statement can be integrity protected by the digital signature of the issuer and may also specify a validity time. Furthermore, mechanisms to exchange such attribute statements (and other security tokens) are provided by standards such as WS-Trust, WS-Security, and the SAML protocols. For an overview of these standards, as well as XACML, the reader is referred to [5,6].

As may be noticed from the previous discussion, the SOA standards required to realize

[1] Identity based access control, where privileges are assigned to individual subjects, is generally not well suited for use in cross-domain scenarios due to its limited scalability and potentially high administrative cost. Role based access control [13] mitigates these problems by assigning privileges to roles, where a subject may be assigned a set of roles. That way, the access privileges of a subject are determined by its set of roles. Even when not using identity based access control, unique/traceable identifiers may still be required for logging/auditing purposes.

interoperable cross-domain access control are readily available. The question however is to what extent current solutions, intended for use in civilian applications, are suitable for use in military systems. That is a topic of this report. In particular, we will consider two issues that are of particular concern in military systems, namely availability and assurance.

In addition to protecting the confidentiality and integrity of the protected resource, a main purpose of access control is to ensure the availability of the resource by preventing unauthorized users from depleting its capacity. In that sense, access control is counter-productive if it prevents a legitimate user from accessing a resource.

Clearly, the communication pattern of the access control architecture introduces additional connectivity dependencies in a system. Thus, while the availability of a resource should ideally only rely on the availability of the resource itself and the availability of the communication channel(s) between the subject and the resource, the availability of the resource is also dependent on the availability of the access control architecture and its dependencies.

In a cross-domain scenario the subject (i.e., the requester) and the resource are being administered by different domains. This imposes an additional challenge with regard to ensuring that the attributes of the subject are available in order to control access to the resource. Because an administrative domain may include multiple networks, the subject, the resource, and the attribute provider may all reside within different networks, making them more susceptible to connectivity disruptions. This risk of connectivity disruptions is further increased by the existence of mobile users and devices.

Although this may not be a significant concern in stable networks with strong connectivity and little at stake, this is a critical issue in some military systems (e.g., tactical networks) where connectivity disruptions must be expected. Timely access to required information and services is often of high importance in such systems, and should therefore to as little extent as possible be degraded. Furthermore, military systems also poses some special characteristics in that some domains are only interconnected through one-way diodes and that some nodes may be required to maintain radio silence. Hence, a solution for cross-domain access control in a military SOA should preferably be able to operate under these conditions.

Another area where military systems differ from typical civilian Internet applications is with regard to the requirement for assuring that the security mechanisms work correctly (i.e., assurance). The previously mentioned SOA security standards are typically implemented as part of large software libraries, which may not provide sufficient assurance for military use.

The rest of this report is organized as follows. In the next section we consider the typical communication patterns of cross-domain access control solutions in service-oriented architectures, and find that these approaches may not be directly applicable for use in many military systems due to their connectivity requirements.

Then, in Section 3, we consider how the trustworthiness of SOA access control solutions can be increased to meet the assurance requirements of military systems.

Finally, in Section 4, we propose an alternative approach in order to mitigate the problems identified in sections 2 and 3.

2 SOA access-control patterns

We will refer to the entity enforcing access control as the control point.[2] Clearly, to be effective, the control point needs to be positioned in such a way that the resource can not be accessed without going through the control point (which may potentially be integrated with the resource). The resource could for instance be a Web service or a network domain.

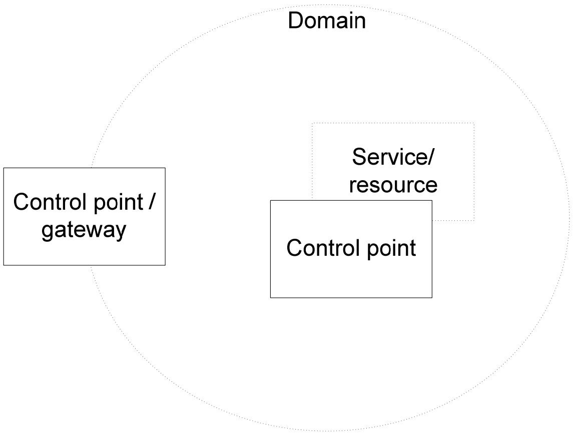

Figure 2.1 Two control points. One is located at the domain border/gateway and the other is collocated with the service/resource.

In a cross-domain scenario, there may be control points both at the resource and at the gateway of the resource's domain as shown in Figure 2.1. If access control is performed both at the domain gateway and at the resource, these access controls may be performed according to different policies or according to identical policies. In the prior case, the control point at the domain gateway might perform access control to the domain (i.e., with the domain as the resource), while the control point at the resource would perform access control to the resource. In the latter case, where the same policy is applied at both control points, early control may be performed at the domain gateway thereby preventing unnecessary consumption of network resources in the domain of the resource. This may be of particular benefit in domains with limited network resources (e.g., tactical networks). As may be noticed, it requires that the access control policy for the resource (and if applicable the attributes of the resource) is also made available to the domain gateway control point.

[2] A control point is assumed to provide the functionality provided by a policy enforcement point (PEP) and a (collocated or remote) policy decision point (PDP) combined. For availability and performance reasons we recommend that the PEP and PDP are collocated unless there are particular reasons otherwise.

Positioning a control point in an intermediary domain or within the domain of the subject is conceptually different. This could be motivated by the need for network resource control in those domains, and in the latter case also for client security reasons (i.e., restricting the subject from accessing potentially security compromising resources). Although many of the same principles are applicable to such scenarios as well, they are not the focus of the following discussions. When not stated explicitly otherwise, however, the following discussions in this document are independent of whether the control point is positioned at the gateway of the resource domain and/or at the resource (or anywhere in between).

2.1 Control point retrieves subject attributes

One possible strategy is to have the control point retrieve the subject attributes from a repository (potentially implemented as a security token service), as shown in Figure 2.2. In such a case, the repository may either be collocated/integrated with the access control point or be a separate (distributed or centralized) repository. Integrating the attribute repository with each access control point has the advantage of ensuring the availability of the attribute repository when performing access control. In a coalition scenario, however, this would require individual user information from each domain (e.g., nation) to be distributed to the control points of all the other domains (i.e., at least to the ones where access is expected to be gained), thereby imposing significant scalability and management issues. Such an approach to an extent eliminates the advantages of having an attribute/role based access control, as individual user information would still need to be widely distributed. This would particularly be a concern when control points are located in networks with low bandwidth.

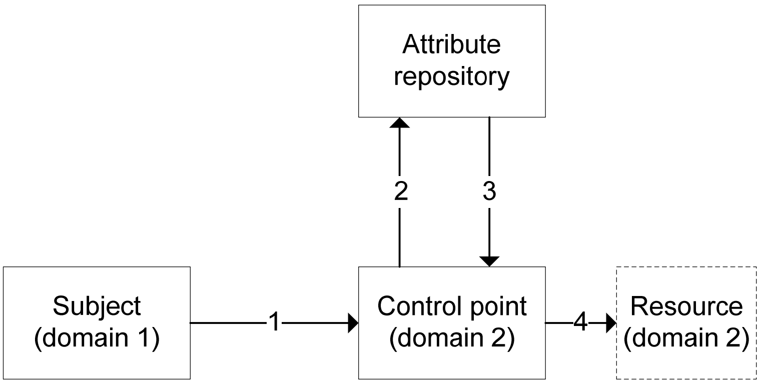

Figure 2.2 The control point retrieves the attributes of the subject upon receiving the access request from the subject.

An alternative is to have the access control points retrieve the attributes of the requester from a centralized location. Using this scheme, each domain would maintain a repository of its users, providing their attributes to access control points upon request. Data replication (i.e., either between redundant repositories or within a distributed repository implementation) would still be required for redundancy, but all attribute management would remain within the originating domain of the user.

A disadvantage with this scheme compared to the previous one is that it requires a two-way channel between the domains, and as such will not work when the domain of the subject only has a one-way channel to the domain of the resource. Also, it would not work if the control point (resource) is required to maintain radio silence.

Because the control point would always obtain the fresh attributes of the subject, this scheme ensures that there is a low delay before changes to the attributes of a subject takes effect. On the other hand, because the control point needs to wait for the reply from the attribute repository, it incurs additional delay to the processing of access requests. Furthermore, such an approach may also generate a significant amount of network traffic between control points and attribute providers, which can be triggered by users unauthorized to access the resource.

Not the least, such a scheme is highly vulnerable to connectivity disruptions between the attribute provider and the control point, in which case the requester/subject would be unable to access the resource.

While a combination of the two above schemes is possible, e.g., by synchronizing subject attributes between repositories within each domain, such a scheme would also require extensive pre-distribution of user information and be subject to many of the same problems.

Thus, although such a strategy may be suitable for use in systems with strong connectivity, it is not appropriate for use in less well connected networks such as tactical networks.

2.2 Subject presents attributes to control point

In order to ensure the availability of subject attributes to control points, the alternative besides pre-distributing user information between domains is to have subjects carry one or more attribute statements specifying their attributes. A subject then presents its attribute statement(s) to the control point when making an access request. In order for access control decisions to be based upon such statements, the following requirements must be fulfilled:

- The attribute statement(s) must have been issued or vouched for by an entity trusted by the relying control point.
- The integrity of the statement(s) must be ensured.
- The association between the statement(s) and the subject must be verifiable.
- The information within the statement(s) must be trusted to still be valid (freshness).

The first requirement deals with the trust relationship between the control point and the issuer of the attribute statement, i.e., a control point is only going to accept attribute statements from issuers which it trusts. This trust relationship can be expressed in the policy of the control point, and is not necessarily an either or relationship. For instance, a control point may trust an issuer to issue only certain types of attribute statements. Also, two identical attribute statements issued by two different issuers could potentially result in different access rights. Furthermore, a control point may potentially also accept attribute statements from issuers which it only has an indirect trust relationship. This would be the case if the control point allows access based on an attribute statement issued by an issuer which it does not trust directly, but which is again trusted by another issuer trusted by the control point to perform such delegation/indirection of issuer rights.

As may be noticed, managing trust relationships can become fairly complex even when not considering the more non-technical (e.g., organizational or political) issues. For our purpose, however, it is sufficient that these relationships can be expressed in the policies of the control point(s) and the statement issuers. Specifically, the trust relationships of a control point may be expressed using the unique identifiers of the trusted statement issuers or by specifying the attributes that trusted issuers are to have. The first approach is the simpler approach when the control point is only to accept statements from a single issuer. However, as additional issuers are to be trusted the attribute based approach may be preferable. When assigning attributes to an issuer, these attributes may either be assigned by another (potentially higher level) issuer or by the control point mapping issuer identities to attributes.

The integrity of the attribute statements must also be ensured. As the statements are not transferred directly from the issuer to the control point, the integrity protection must be associated with the statements themselves and not with the transmitting channel. A typical solution, as provided for in SAML, is to have the issuer sign each attribute statement, enabling the control point to verify that the statement was in fact issued by the claimed issuer, and that the statement has not been modified afterwards.

Even though the attribute statement itself may be valid, it must also be verified that the statement does in fact apply to the subject at hand. There are several ways in which this can be done. One possibility is to specify the identity of the subject in the statement, and have the subject authenticate itself to the control point using a separate authentication mechanism. This requires that the control point and the subject share a common authentication mechanism. Another option is that the subject proves knowledge of a secret specified in the attribute statement. Typically, the attribute statement would specify a public key of the subject, and the subject would be required to prove knowledge of the corresponding private key by signing the message (containing the attribute statement) using that key. In SAML, this is known as holder-of-key subject confirmation.

The last requirement concerns the freshness/validity of the information within an attribute statement. To ensure the validity of the information within attribute statements, and to limit the potential misuse of attribute statements (e.g., if the associated private key has been compromised), it is preferable that attribute statements have a short validity time. On the other hand, from an availability perspective, it is desirable that attribute statements have a long validity time to prevent denial of service in situations where the subject is unable to renew its attribute statement (e.g., due to a loss of connection with the issuer). In such a situation the subject could effectively be denied access to all services, even when the subject has a valid connection to the service itself and in principle should be allowed to access the service. Thus, the requirement for attribute statement freshness is in conflict with the requirement for high availability of services. Although one might argue that a revocation mechanism could be used to resolve this conflict, such a revocation mechanism would be subject to much of the same issue. Thus, as both freshness and availability are of critical importance, we will consider the architectural impacts on freshness and availability in the following discussion.

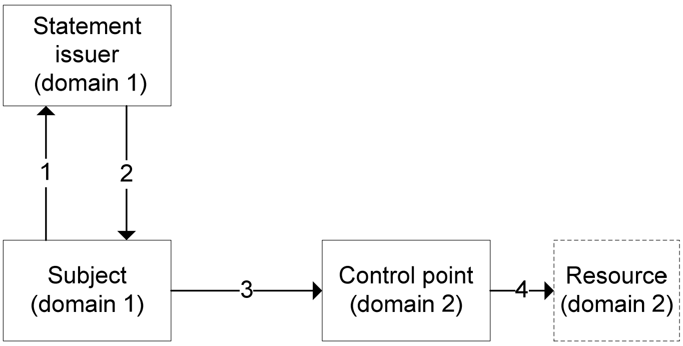

Figure 2.3 The subject obtains an attribute statement from the statement issuer within its own domain and uses this to access the resource.

In its most basic form, the subject would obtain an attribute statement from a statement issuer (e.g., a security token service) within its own domain and then use this attribute statement for accessing the service within the foreign domain, as illustrated in Figure 2.3. An advantage with this scheme is that the client only has to relate to the statement issuer(s) within its own domain, and that it would work even when the domain of the client only has a one-way channel to the domain of the resource. On the other hand, the control point(s) in the foreign domain needs to trust the statement issuer(s) in the domain of the subject. Considering that the number of domains may potentially be high, and that each domain will typically have many control points, there may be significant scalability issues with such an approach. This is particularly the case when considering that control points may need to be embedded on a variety of platforms, including handheld devices and sensor devices.

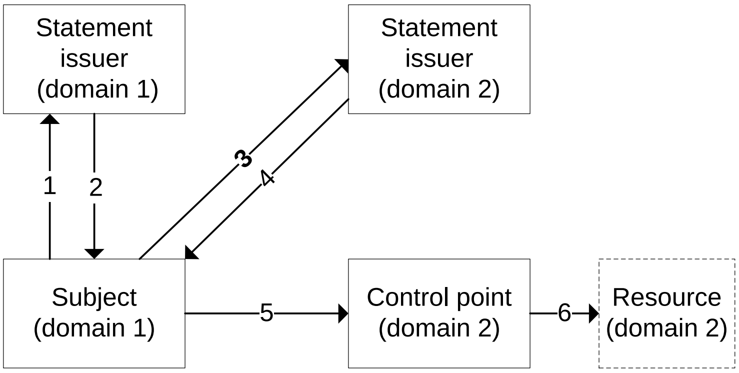

Figure 2.4 A federated approach where the subject obtains an attribute statement from the statement issuer within its own domain, which is then used to obtain an attribute statement from the statement issuer within the domain of the resource.

A typical approach to solve this problem, e.g., using SAML and/or WS-Trust/Federation, is to use a federated solution as shown in Figure 2.4. Using such an approach, the subject still needs to obtain an attribute statement from an issuer within its own domain. However, in order to access a resource in another domain it must have an attribute statement issued within that domain. More specifically, it would use the attribute statement issued within its own domain to obtain an attribute statement within the foreign domain. The implication of this is that complexity is moved from the control points to the attribute statement issuers. That is, each control point now only has to trust the issuer(s) within its own domain, while the issuers have to enforce the trust relationships with the foreign domains. Given that the number of statement issuers within a domain would typically be much lower than the number of control points, this is an advantage. This is especially the case when considering that the statement issuers are typically hosted on more powerful and managed devices.

It may be noticed that this approach adds some additional complexity to the subject/client, as it now has to relate to statement issuers within multiple domains. Although the increased complexity of the subject/client may not itself be an issue, this also has some impact with regard to availability. More specifically, it requires the subject to have access to both the statement issuer within its own domain and the statement issuer of the foreign domain before it is able to access a service within the foreign domain. Hence, additional connectivity dependencies are introduced, in particular when the statement issuers are not located within the local networks of the resource and subject. This may not be a significant issue for (non-critical) well connected systems, such as typical services provided through the Internet. However, it does give reason for concern when considering a military system, especially if cross domain interoperability and information exchange is to be provided at the tactical level.

As the statement issuers are highly security critical services, replicating these services close to the end-user does not provide a good solution to this availability problem. Thus, it implies that the attribute statements need to have a sufficiently long validity time so that attribute statements can be issued well in advance to be resilient to a temporary loss of connection. The described model is however intended for obtaining the attribute statements just prior to accessing the resource, and such an alternative approach may impose a requirement for a cross-domain attribute statement revocation mechanism, which is undesirable. Hence, while such a solution may be suitable for systems with strong connectivity, it is not appropriate for systems with weaker connectivity. Also, such a solution assumes that there is a two-way channel between the domains.

There are also some variations of the above schemes, as described in [7]. However, these alternatives are not motivated by the issues discussed here and are associated with the same problems with regard to use in military systems.

3 Assurance

The trustworthiness of statement issuers, control points, and client implementations (i.e., subjects) are all critical to the overall trustworthiness of an access control solution. We will therefore now briefly discuss the particular issues of each of these entities with regard to assurance.

3.1 Client/subject implementation

The most critical aspect of the subject (i.e., client) implementation is the security of the authentication mechanism, used to prove the association between the subject and the attribute statement. If this mechanism was to be compromised, it could enable an imposer to act as the subject taking advantage of its access privileges. Hence, given that a key-pair specified within the attribute statement is used to prove this association (i.e., the holder-of-key method), the exposure of the private key would compromise the system. As the Web service standards used in these solutions are generally provided as part of large libraries, there would be low assurance that this could not happen.

There are alternative ways to approach this. If a trusted authentication mechanism is available to the subject at another layer (e.g., through the use of an IPSec device), utilizing this mechanism might provide the simplest solution. The attribute statement would then simply specify the identity of the subject to which it applies.

If one are to use the holder-of-key method, the private key needs to be protected. In [8] it is demonstrated that a Web service platform can be split into two parts, a smaller trusted part and an untrusted part. The amount of functionality provided by the trusted part is quite extensive in [8], and the separation between the trusted and untrusted part is provided by running the untrusted part as an unprivileged user while the trusted part is being run as a privileged superuser. While this might give the trusted part excessive privileges, a somewhat similar strategy may be used. Specifically, only the signature mechanism (including the private key) would need to be considered as a trusted part for our purpose, and as such be separated from the rest of the system. One way to achieve such separation is by utilizing specific purpose hardware for creating the signature (e.g., a smartcard or trusted platform module [9]), where the key remains protected. Alternatively, strong separation can be provided in software using a solution like multiple independent levels of security (MILS) [10] [11].

It should be noted though that the assurance requirements for the client/subject implementation depends on the privileges of that subject (as a subject with no privileges poses little risk). In a cross-domain scenario, however, it may be difficult to assess exactly what resources a given subject would be able to access.

3.2 Statement issuers

Because the compromise of a statement issuer could enable illegal access to a wide range of systems, the statement issuers are the most critical entities with regard to the security of the discussed access control solutions. Consequently, the private key of a statement issuer also needs to be protected from exposal as discussed above. However, this is not sufficient to prevent someone who is able to take control of the statement issuer (e.g., by installing a backdoor) from issuing false attribute statements. Such false attribute statements might give illegal access to all the resources that subjects from that domain may be allowed to access (i.e., the resources within the domain itself in addition to potential resources within other domains). Based on this, statement issuers should be well protected.

However, existing cross-domain SOA solutions typically take quite a different approach here. If we consider the federation solution in Figure 2.4, subjects from all the domains must be able to access the statement issuer within the domain of the resource. Hence, this highly security critical service is widely exposed.

The highly security critical nature of this service is due to the dynamic creation of attribute statements upon request. Because the validity/expiration time of a SAML assertion is integrity protected by the same signature as the remainder of the SAML assertion, it is not possible to extend the validity time of an assertion without breaking the signature. Also, because SAML has no revocation model, SAML assertions are recommended to have a short validity time. Hence, a security token service is typically not just a repository of SAML assertions, but needs to be able to create assertions (with an appropriate validity time). This makes the security token service highly security critical, as a compromised security token service can be used to forge any type of assertion. Hence, it would be better if the exposed security token service could operate more as a repository of SAML assertions.

In order to make this possible, one might create a new type of SAML condition. A SAML condition is a SAML construct that places constraints on the use of a SAML assertion, and SAML provides an extension for defining new condition types. The new condition type would specify that a co-signature is required for the assertion to be valid, and the public key to be used to verify that co-signature. Such a co-signature would then also be required have a specified validity time, which should be within (typically shorter than) the validity time of the assertion.

The co-signature may be implemented using a SAML authorization statement with a specified validity time (again within the validity time of the attribute statement). This authorization statement would then function as the co-signature and authorize the subject to use its attribute statement for requesting access. The advantage with using an authorization statement for this is that an authorization statement may contain or reference another SAML assertion. That way the attribute statement assertion can be included within the authorization statement assertion.

This way, the creation of the attribute statements could be moved to a separate statement issuer. This issuer would issue SAML assertions, with a relatively long validity time, for all the subjects within its domain. These assertions would then be sent to a statement provider (e.g., using a one-way channel). When issuing an assertion to a subject, the statement provider specifies the validity time and creates the co-signature, making the assertion valid for use in that period. This way, it would not be possible to use a compromised security token service (i.e., statement provider) to issue additional privileges. Also, a statement provider is only able to renew/validate an attribute statement within the base validity time specified by the statement issuer.

However, if we again consider the federation solution from Figure 2.4, this solution depends on the security token services (i.e., statement issuers) being able to create attribute statements dynamically. This is because the statement issuer in the domain of the resource is required to create a new attribute statement for the subject. Although one might think that this could be solved by having the statement issuer in the foreign domain simply apply a co-signature to the subject's original attribute statement, this is not the case. That is because the control point has no relation to the statement issuer in the subject domain, and would therefore not be able to verify whether the original attribute statement was issued by a proper issuer or not. Hence, such a scheme would nevertheless enable a compromised statement issuer within the domain of the resource to create false attribute statements granting excessive privileges.

3.3 Control points

The criticality of a control point depends on the criticality of the protected resource. The correct operation of a control point depends on trusted delivery of the public key used for verifying the signature of attribute statements, as well as the integrity (and potentially also freshness) of the access control policies. A control point is also depending on having a clock that is sufficiently synchronized with the clock of the issuer of the statements which it relies on. To provide higher assurance, a control point may also be implemented using MILS to partition the system into smaller parts and ensure non-bypassability, or be implemented as a custom hardware device.

3.4 Message security

The integrity of the exchanged messages is typically protected using a digital signature, which may then also be used for holder-of-key subject confirmation. Furthermore, a mechanism to avoid replay attacks should also be included within messages. In order for such a mechanism to work in the presence of one-way channels, a challenge-response type of mechanism can not be used between subjects and control points (i.e., resources). Typically, a timestamp (as provided for by WS-Security [12]) in combination with a specification of the intended recipient may be used. That way the recipient can cache the timestamps of messages received from a given subject for the timeframe for which the messages would be accepted. In store-and-forward networks with high delays, the timeframe for which a message should be accepted would then need to be accordingly long. If this could result in a very large cache, an alternative strategy where only the newest timestamp from each subject is kept may be used. However, this could cause legitimate messages to be discarded as duplicates if messages are reordered during transport. Finally, confidentiality/privacy requirements might also require the assertion or certain attributes to be encrypted.

4 Alternative solution

In order to mitigate the problems discussed, in sections 2 and 3, we here propose an alternative approach. Aiming to reduce the connectivity dependencies of the access control architecture, the subject only needs to relate with the statement issuer within its own domain. At the same time, control points are only required to have a direct trust relationship with the statement issuer within their own domain. Furthermore, the proposed solution does not depend on dynamic creation of attribute statements upon request, enabling the security critical statement issuers to be better protected.

4.1 Description

The basic idea of the proposed approach is illustrated in Figure 4.1. It is based on having the domain of the resource issue an attribute statement to the statement issuer in the domain of the subject (i.e., step one in the figure). This attribute statement serves the purpose to grant the statement issuer in domain 1 (i.e., the domain of the subject) with specific issuer rights within domain 2 (i.e., within the domain of the resource). This may happen in advance and independent of the subject requesting an attribute statement.

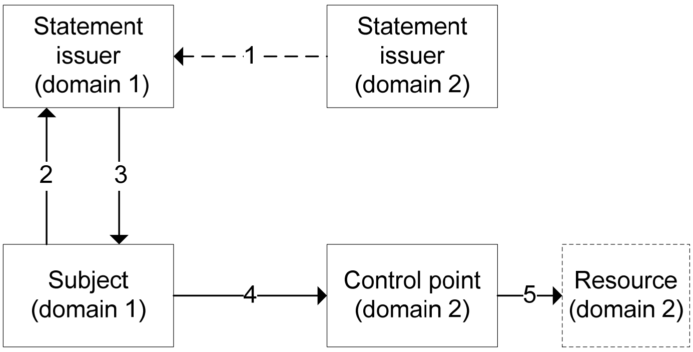

Figure 4.1 Domain 2 issues an attribute statement to the statement issuer in domain 1, granting it certain issuer rights in domain 2. The subject uses this attribute statement together with its own attribute statement to request access within domain 2.

Now, when the subject requests to have an attribute statement issued by its statement issuer (step 2), the statement issuer returns (step 3) an attribute statement for the subject and includes its own attribute statement issued by domain 2 (i.e., the one it received in step 1). Then, when the subject is to access the resource, it includes both its own attribute statement and the attribute statement of the statement issuer in the access request. The control point then use both received attribute statements in order to decide the access request.

Let us take a closer look at the chain of trust under this model. First, the statement issuer in domain 2 is trusted by the control point in domain 2 (and the control point has access to the statement issuer's public key). The attribute statement of the statement issuer in domain 1 is issued and signed by the statement issuer in domain 2. Furthermore, this attribute statement specifies/contains the public key of the statement issuer in domain 1. The corresponding private key is then used for signing the attribute statement of the subject. This way, there is established a chain of trust so that the control point can trust the attribute statement of the subject. Finally, the attribute statement of the subject also includes the public key of the subject, so that the subject can prove its ownership of the attribute statement (by signing the request message using the corresponding private key).

The attribute statement issued to the statement issuer in domain 1, by domain 2, also requires a little more detailed explanation. There are several ways in which the attributes within this statement can be used to grant issuer rights within domain 2. One alternative is that this statement specifies the exact attributes that the statement issuer in domain 1 is allowed to grant within domain 2. In this case, the resulting attributes used by the control point for deciding an access request would be obtained using an AND like operation on the attribute statements of the issuer and the subject.

A more streamlined approach, however, is obtained by including the attributes within the issuer's attribute statement as environmental (i.e., contextual) attributes when evaluating the access request. That way, the attributes of the issuer define the context for the evaluation of the access request of the subject. This could for instance be used to identify according to which policy the access request is to be evaluated. In any case, the final access decision is made by domain 2, and the foreign issuer is not able to grant excessive attributes/privileges within domain 2.

In order to provide additional protection of the statement issuers, the statement issuers may be split in two as discussed in Section 3.2. The resulting architecture, where statement providers are separated from statement issuers, is shown in Figure 4*.2*. The statement issuer within domain 2 issues an attribute statement for the statement issuer in domain 1 (step 1a). Because the statement issuer in domain 1 does not require possession of this attribute statement, it may be sent directly to the statement provider of domain 1 (potentially using a one way channel). This way, the statement issuers can remain highly protected.

The statement issuer within domain 1 issues attribute statements for the subjects within its domain and sends these attribute statements (potentially using a one-way channel) to the statement provider (step 1b). These steps (1a and 1b) happen in advance and independent of the subject requesting an attribute statement.

When the subject requests its attribute statement (step 2), the statement provider applies its co-signature, as explained in Section 3.2, making the attribute statement valid for a given period. The statement provider then returns the attribute statement to the subject (step 3), together with the attribute statement of the issuer. As before, these attribute statements may then be used by the subject to access the resource.

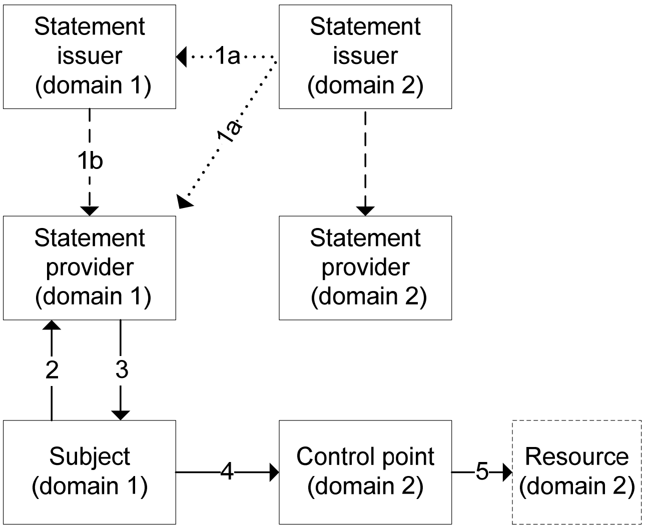
Statement issuer (domain 1)
Statement issuer (domain 2)
1a
1a
1b
Statement provider (domain 1)
Statement provider (domain 2)
2
3
Subject (domain 1)
4
Control point (domain 2)
5
Resource (domain 2)

Figure 4.2 The statement issuers have been separated from the statement providers. The attribute statement issued by the statement issuer in domain 2 to the statement issuer in domain 1 (step 1a) may alternatively be sent directly to the statement provider of domain 1 (or be distributed internally within domain 2).

4.2 Discussion

Let us examine the implications of this proposed solution in more detail. With regard to availability, it can be noticed that the subject is only required to interact with a single statement issuer/provider. Although the subject is also depending on this issuer/provider being able to obtain an attribute statement from the foreign domain, we argue that this is of less concern from an availability point of view. This is because such attribute statements between issuers can be issued with a longer validity time than for typical subjects (e.g., end-users), and hence be exchanged well in advance.

There are several reasons for this. For one thing, the attributes of an issuer (i.e., domain) can be expected to be relatively stable, because the relationships between domains (e.g., the nations within a coalition) are not likely to change frequently. Furthermore, statement issuers are to be well protected services whose compromise would be a rare exception. Considering the potentially severe consequences, one nevertheless needs to be able to handle such an exception but this can be handled internally within the domain. More specifically, under this model, an attribute statement issued to domain 1 by domain 2 is only valid within domain 2. Hence, there is no need for a cross-domain revocation mechanism in order to revoke such an attribute statement. Revocation may therefore effectively be performed by updating the access control policy/policies within the domain or using some other revocation mechanism internal to the domain.[3] It may also be noticed that a new attribute statement, potentially with a shorter validity time than the original, may be issued between the domains at any time. This can be used if additional issuer rights are to be temporarily granted.

[3] If an X.509 infrastructure is utilized within the resource domain, the attributes of foreign statement issuers may potentially be expressed using X.509 attribute certificates [14] and be distributed internally within the domain of the resource.

With regard to scalability, the proposed solution provides comparable scalability to the most scalable of the discussed existing solutions (i.e., the federation solution in Figure 2.4). The inter-domain trust patterns of these two solutions are in fact identical and it is as such only the communication/exchange patterns and the use of the statement issuer's attributes during policy evaluation that makes up the difference.

We now consider the case when there is only a one-way connection from the domain of the subject to the domain of the resource (the other way around is not of interest as it would not be possible for the subject to request access to the resource). Although the subject will not receive any response from the service, this can be used to submit data or controlling commands to a resource. Given that the statement issuer is located in a partition separate from the rest of domain 2, this partition could be connected to the rest of domain 2 as well as the statement provider/issuer of the other domain(s) through one-way channels. This assumes that the attribute statements issued by the statement issuer of domain 2 do not have a confidentiality classification higher than the connected domains, or that a filter/guard/review mechanism ensures that any confidential attribute statements are not released. Alternatively, the attribute statement of the statement issuer in domain 1 may be made available to the control points within domain 2 directly, using an internal mechanism within that domain. In any case, the statement issuer in the resource domain is able to issue an attribute statement for the statement issuer within the subject domain, enabling the subject to access the resource through a one-way diode. Furthermore, the control point is not required to send any messages, enabling the resource to maintain radio silence.

Although the subject is now only required to interact with one statement issuer, we see that there is still a tradeoff between ensuring availability (i.e., access to services) and the validity/freshness of the subject's attributes. This is a problem that can not be avoided completely. One way to make the best of this though could be through the use of adaptive policies. This way the freshness requirement on attribute statements could be adapted depending on the connectivity characteristics of the network, the current threat/risk level (both considered as environment attributes during policy evaluation), and the criticality characteristics of the resource (e.g., considering the importance of its availability vs. confidentiality). That way, poor network connectivity and a low threat level might allow for the acceptance of relatively old attribute statements, while good network connectivity and a high threat level would mandate the use of fresh attribute statements.

Also, because statement providers are less security critical than statement issuers, a statement issuer may have multiple statement providers that can be positioned closer to the end-users (i.e., the subjects). This way, the probability of a valid connection between a subject and a statement provider is increased. At the same time, the connection between statement providers and statement issuers can be made more resilient to connectivity disruptions by having a longer base validity time on the attribute statements (recall that such attribute statements are not usable without the co-signature of the statement provider). Still, unless a separate revocation mechanism is available (e.g., through policy updates), the base validity time needs to be kept short enough to prevent a compromised statement provider from co-signing attribute statements for a prolonged period.

Finally, it may be noticed that the discussed solution does not depend on a public key infrastructure (PKI). However, the existence of a public key infrastructure could be advantageous for instance to distribute the statement issuers' public keys and/or attribute statements (i.e., if using X.509 attribute certificates instead of SAML attribute statements for this purpose).

5 Summary

In this report we have considered typical solutions for cross-domain access control in service-oriented architectures and found that they are not directly applicable to many military systems, because of different requirements for availability and assurance. In particular, the common practice of having security token services dynamically create attribute statements upon request represents a security risk that could result in the compromise of the access control system. Furthermore, the communication dependencies created by typical SOA access control solutions may be deterrent for the availability of a system.

To mitigate these problems we have proposed an alternative approach with lower connectivity requirements and that does not require attribute statements to be dynamically created. This alternative approach can be implemented using existing SOA standards, and can be used even when the subject only has a one-way channel to the resource (e.g., because the domains are interconnected by a one-way diode or because the resource is required to maintain radio silence).

Still, a tradeoff between availability and ensuring the validity/freshness of attribute statements is not to be avoided. We suggest that adaptive access control policies may be used to provide the best tradeoff. More specifically, the connectivity characteristics of the network and the risk/threat level (in addition to the criticality of the specific resource) can be used as environmental attributes during policy evaluation to determine the maximum accepted age of attribute statements. As such, the proposed approach does not aim to provide a final solution, but may serve to provide direction for further research and investigation.

References

[1] T. Moses, "eXtensible Access Control Markup Language (XACML) Version 2.0," OASIS Standard, 2005.

[2] E. Yuan and J. Tong, "Attribute Based Access Control (ABAC) for Web Services," in *Proc. IEEE International Conference on Web Services*, 2005.

[3] D. E. Bell, "Looking Back at the Bell-La Padula Model," in *Proc. 21st Annual Computer Security Applications Conference*, 2005, pp. 337-351.

[4] S. Cantor, J. Kemp, R. Philpott, and E. Maler, "Assertions and Protocols for the OASIS Security Assertion Markup Language (SAML) V2.0," OASIS Standard, 2005.

[5] N. A. Nordbotten, "XML and Web Services Security Standards," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 4-21, 2009.

[6] N. A. Nordbotten, "XML and Web Services Security," FFI-rapport 2008/00413, 2008.

[7] J. Rodriguez and J. Klug, "Federated Identity Patterns in a Service-Oriented World," *The Architecture Journal*, no. 16, pp. 6-11, Jul. 2008.

[8] J. Wei, L. Singaravelu, and C. Pu, "A Secure Information Flow Architecture for Web Service Platforms," *IEEE Transactions on Services Computing*, vol. 1, no. 2, pp. 75-87, 2008.

[9] Trusted Computing Group, "TPM Main Specification Level 2 Version 1.2, Revision 103," 2006/2007.

[10] T. Gjertsen and N. A. Nordbotten, "Multiple independent levels of security (MILS)," FFI-rapport 2008/01999, 2008.

[11] T. Gjertsen and N. A. Nordbotten, "Military opeartional systems in field - multiple levels of security," FFI-rapport 2009/01137, 2009.

[12] A. Nadalin, C. Kaler, R. Monzillo, and P. Hallam-Baker, "Web Services Security: SOAP Message Security 1.1 (WS-Security 2004)," OASIS Standard Specification, 2006.

[13] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman, "Role-Based Access Control Models," *Computer*, vol. 29, no. 2, pp. 38-47, 1996.

[14] D. W. Chadwick, A. Otenko, and E. Ball, "Role-Based Access Control With X.509 Attribute Certificates," *IEEE Internet Computing*, pp. 62-69, Mar. 2003.