Consiglio Nazionale delle Ricerche

Access control: AAAA where the first two As stand for Ability Authentication

F. Dianda, F. Giuntini, F. Martinelli, A. Vaccarelli

IIT TR-05/2004

Technical report

Giugno 2004

Istituto di Informatica e Telematica

Access control: AAAA where the first two As stand for Ability Authentication*

Fabio Dianda, Filippo Giuntini, Fabio Martinelli, Anna Vaccarelli
Istituto di Informatica e Telematica C.N.R., Pisa, Italy
{Fabio.dianda, Filippo.Giuntini, Fabio.Martinelli, Anna.Vaccarelli}@iit.cnr.it

Abstract

We investigate the notion of authentication procedure. We show how this can be split in several phases: establishing that an ability can be performed, deducing a certain information from such an ability, witnessing the authenticity of the information deduced from the ability. In our framework, ability means that that one is able to perform role in a protocol. We argue that the concept of ability authentication is more general that the one of identity authentication, that is commonly used in AAA systems. We then develop an architecture for managing abilities and we show some application scenarios.

1 Motivation

As noted in [1], authentication is a process, that is intended to establish, with a sufficient degree of certainty, the truth of some claim. Much of model presented until now have been focused on authenticating claims that permit to distinguish between the individuals in a community (identifiers), or claims associated with property of an individual (attributes); in other words the focus has been on identity authentication, which could be defined as the process of establishing an understood level of confidence that an identifier or an attribute refers to an identity [1]. An entity participating in this process can play two roles:

- Claimant: who (user, device, etc.) states, directly (ex: password) or indirectly (ex: fingerprint), an information by performing an authentication protocol.
- Verifier: who has the task to verify the veracity of a claimed information.

*Work partially supported by MIUR project "Tecniche e strumenti software per l'analisi della sicurezza delle comunicazioni in applicazioni telematiche di interesse economico e sociale"; by CNR project "Strumenti, ambienti ed applicazioni innovative per la società dell' informazione" and finally by CREATE-NET "Quality of Protection".

Generalizing what is defined about identity, it is possible to note that to be authenticated, a claimant perform a particular, often predefined protocol, which uses a parameter, the so-called authentication factor. This factor is based on a property or a characteristic associated with the identity and could be classified in:

- Something you know
- Something you have
- Something you are
- Something you do
- Something you come from

Note that generally, the protocol to be used is tightly coupled with the information to assert: to gain access to a resource located on your lan, it could be requested to authenticate by means of an ip address belonging to a particular address space, but if the user is connected from a different location, for example he/she is using his/her isp home account, not only this method could not be used, but he/she could not authenticate or be authenticated too.

As illustrated in [9], one of the main shift in today virtual organizations is to rely on users management performed by a third party, which could be nearer and know better the users that request access to remote resources. This approach is indicated by the industry as federated identity management, to group the processes and the supporting infrastructures for the creation, maintenance and use of digital identities within a legal and policy context [15]. These efforts promoted by various initiatives [16], [18], [19], shows how authentication and authorization are two different processes, that can be separated in order to take advantages of local knowledge of the users for authentication, of the resources for authorization.

Although identity is a fundamental concept in building and establishing trust between parties, it is not always useful to authenticate it, or better, in some environments it could be more useful to establish the truth of other type of claims: consider for example a typical e-commerce scenario, where a merchant leverages the functions performed by a payment gateway to manage the credit card payments. To fulfill an order the buyer has to perform a protocol in order to obtain the authentication of its payment promise by the payment gateway, who in ultimate, witness to the merchant this capacity of the buyer to authorize the money transfer between the accounts. The merchant is not much interested in knowing the name of the person printed on the credit card, it is much more interested in knowing if the payment has been authorized by the gateway, or in the other words if the payment promise received is authentic.

This simplified example shows once again that while identification (the process of using claims to infer who the user is), authentication (the process of establishing the truth of some claims) and authorization (the process of deciding what the authenticated subject ought to be allowed to do) are three distinct concepts, but they have to be interrelated in a convenient manner, in order to match the requirements of the applications: not all the apps need to know about authenticity of identity, some of them need to know about other types of information.

In this paper we focus on how authentication come from a common factor, ability to perform a protocol; identity authentication is obtained from a subset of these protocols while from others it is possible to obtain authentication of role assigned, authentication of payment order, authentication of an executed step in a business process etc. One of the main reasons that lead us to this study are derived by observing that authentication is not an end result per se, rather than it is a propaedeutic phase that must be accomplished before other phases of the interaction, so it could be said that a "proper" authentication is the first mandatory step in every business process. Establishing what proper means in the phrase above is the subject of the subsequent sections, that contains a description of the framework we introduce, a family of possible software architectures to implement it, and a brief comparison of other similar models.

1.1 The process of authentication

The majority of the authentication protocols assume that the parties involved in an instance of one of them perform both two distinct phases. On the one side, the claimant has to

1. Perform an declaration phase in order to affirm the claims to be authenticated, this phase could be optional and could not require the participation of the user, as for example in ip address authentication.
2. Use in a "proper" way the authentication factor in order to corroborate the phase above.

On the other side the verifier has to check two main aspects of the authentication process:

1. To establish the correctness of an authentication procedure execution instance (the protocol ended correctly).
2. To check the authenticity of the authentication factor used in the protocol.

Following this distinction, we observe that in reality what a claimant demonstrate by means of an authentication protocol could be more significant than the truth of some claims, or better, the latter could be considered a part of a more general and meaningful deduction process.

First, it should be noticed that who authenticates an information is the verifier, by performing a given role V in the authentication protocol; in the same protocol the claimant performs a different role C, or better, *the verifier authenticates an information by means of a claimant demonstration to be able to play the role C in an authentication protocol.*

To stress this behavior we introduce the concept of *Ability Verifier and Witness (AVW), to denote a component that works on behalf of an entity to perform a role V in an authentication procedure P, in order to verify the ability of another entity to perform a different role C in the same protocol.* By checking the two phases mentioned above, an AVW is able to determine the truth of the claims asserted by the other party, in accordance with an authentication protocol correct execution.

This ability could be witness by the AVW to a relying party through the issue of an assertion, which attests that the claimant has been able to perform the authentication process.

Note that each party that needs to authenticate the other should have its own, or rely on, an AVW, for example in a client server model, he AVW should be at least owned by the server in order to authenticate users, but if it is requested to have mutual authentication both client and server should have its own AVW.

For example, let us consider a typical authentication protocol based on something you know, like SSL, considering for the moment the case where each of the parties have a digital certificate. Once it exits a correct protocol execution, in which the parties had executed the phases of the protocol both in their respective role, it is possible to demonstrate each other to be their-selves (identity authentication) but also to demonstrate the ownership and the control the corresponding key pairs (the authentication factor). In other words *authentication, borns from ability to execute a role in a protocol.*

Once it exists a correct instance of an authentication protocol, the AVW can establish the ability of the claimant to perform that procedure; showing this ability has the direct consequence of authenticating the claimed information by means of the used authentication protocol. This deduction could be witness by the AVW to a relying party through the issue of an assertion, in this way it could be possible to use this ability in the time, to present it like a sort of capability to access resources (after an access control policy check for compliance), and in the space, to access resources under a different administrative domain (after the establishment of a proper trust relationships).

1.2 The role of authentication policy

The main concept we argue, is *authentication as a protocol role verification*, from which it derives, trough logical deduction, ability, authentication of claims, and

eventually assertions to be used elsewhere or later on. To lead this deduction phase we introduce a policy, *an authentication policy*, to denote the rules that leads the establishment of the abilities obtainable from a protocol execution.

The role of this policy can be folded in some distinct aspects: *first,* let us note that the performed procedure influences the deduced ability: demonstrating to be able to perform a protocol based on "something you are" is different from performing a protocol based on "something you know"; but also protocols based on the same type of authentication factor demonstrate different abilities, for example using a password based one and a signed nonce-challenge both belong to "something you know", but they show different abilities and, more important, they offer different degree of confidence. As noted in [2], it is useful to classify authentication procedures according to the degree of the offered protection against malfunctioning: the more the protection offered, the more the strength to authenticate. This classification has to be recorded somewhere, we suggest to insert this sort of information in a policy, which expresses the preferences of a particular AVW according to the strength, and the associated risk management approach, of the supported authentication protocols.

Second, to instantiate the classification presented above, we refer to *authentication context*, to denote the procedure, used to an ability. We imagine that different authentication contexts, even if they show different abilities, could be mapped into the same assertion. To clear why, consider for example a mobility scenario, where users need to access informations from different locations, even if they got a strong authentication factor they could find problems in using it in the environment where they are: if to be authenticated it is required to pass a biometric protocol, it is hard to imagine to find a scanner in an Internet cafe, nevertheless the need to access the information remains. Summarizing there is a requirement to support multiple authentication schemes to gain access to an information, but due to the fact that is not always possible or feasible to use the strongest one, it needs to record which one has been to used (the authentication context) and to insert it into the assertion issued by the AVW in order to permit a proper authorization according to a security policy. As a logical consequence, the stronger the mechanism has been used, the powerful is the ability demonstrated, the less restrictive should be the access rights associated with a user.

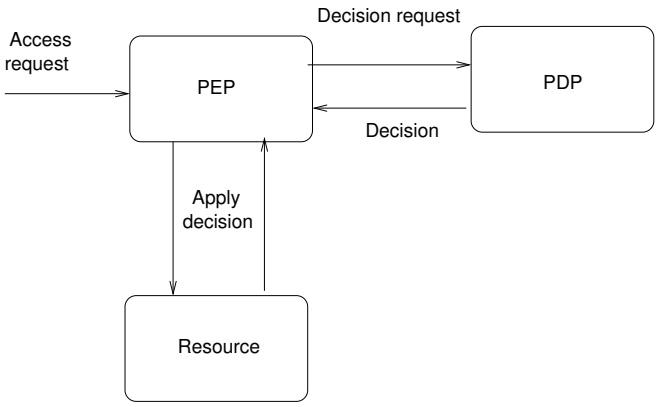
Summarizing the policy should bind together:

- The authentication schemes supported
- The authentication procedure used (the context)
- The deduced ability
- The assertion to be issued

By using an authentication policy we obtain a policy based system, which decouples authentication logic, in our case ability logic from the programs that implement the AVWs, in other words the authentication is going to be programmable.

2 A proposed architecture

One of main area that could take benefits of an ability oriented approach, is authorization; we refer to the model depicted in [4], in which the functionalities of access control are splitted between two entities named Policy Enforcement Point (PEP) and Policy Decision Point (PDP), briefly illustrated in the figure below:



Summarizing PEP function is to mediate the access requests submitted by the users, redirecting the users not already authenticated (in our case those that have not already showed an ability) and enforcing the access decisions established by the PDP according to an access control policy.

In our model, to be habilitated, the user is redirected to an ability verifier, we do not impose an entity to have only one AVW, instead we let the possibility to manage more AVWs that offer an habilitation service to be integrated . The choice to which have to verify the ability procedure is taken by the PEP, according to a part of the access control policy. The general idea is to obtain a flexible authorization, where permissions are related to the ability demonstrated by a subject, the stronger the authentication procedure used the greater the privilege permitted on an object, while at the same time covering the case of strict matching: to grant an access in a given mode it must be executed a fixed procedure with a fixed AVW.

As presented above, if the procedure P is correctly executed, the AVW can deduce, according to its authentication policy, the subject ability to perform the

role R and subsequently it can issue a proper assertion to attest this deduction. The definition of the content of this assertion is considered as an implementation issue, but we imagine a large use of an XML based language, like SAML, to express this information. Note however that [17], has a notation of authentication context, to refer to additional information that a service could require in order to authorize an access request.

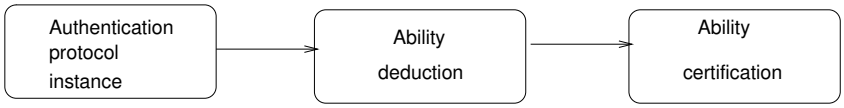
Once a user has been habilitated, he can use the obtained assertion to gain access to the resource, by presenting the received assertion (user push model) or by the delivery of the assertion directly to the PEP (user pull model). In both cases, the PEP performs a different task than redirecting the user, it forwards the request and the assertion submitted to the PDP, in order to permit the check against an access control policy. It should be noted that according the flow described above what is checked against the access control policy are the assertions generated by the AVW, which represents abilities, which represents role checking in a protocol instance.

It should be stressed that like every assertion issued by someone, the assertions issues by the AVW have a temporal validity constraint, so before being authorized, they have to be checked, to determine their validity. We imagine to insert a particular module in the AVW, named *Assertion Checker (AC), that has the task to validate on behalf of another entity the validity of an assertion.* The tasks performed by this entity are inspired by the concept of delegated certificate validation [11], where a third party is demanded to check the validity of a cryptographic credential and of the related trust path (Delegated Path Validation), or only the path discovery (Delegated Path Discovery). In a more general way, it is possible to extend the services offered by this module, imagining that in some protocols, it could be used to validate the authentication factors too, like for example the validity of the certificates used in a SSL based procedure.

Summarizing we imagine an AVW as composed by four components:

- Protocol Role Verifier (PRV): the entity that executes the role of V in an authentication procedure.
- Deduction Engine (DE): the entity that after a correct execution of a protocol instance deduces which ability has been demonstrated, and establishes which assertion has to be issued according to its authentication policy.
- Assertion Issuer (AI): the entity that performs the issue of the assertion to witness the ability deduced. This assertion could be unsigned (like traditional cookies), signed (like some SAML assertions) or encrypted (like Kerberos tickets).
- Assertion Checker (AC): the entity that offers a validity service for the assertions issued by the AVW, and perform or help in the verification of the authentication factors involved in an authentication protocol.

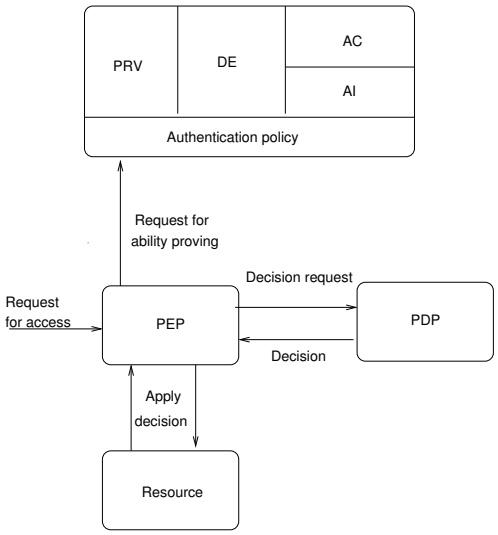
The main flow of the information between these elements is briefly illustrated in the figure below:



The first entity that is involved in an ability procedure is the Protocol Role Verifier, which has the task to perform the authentication procedure instance with the user; it is assumed the existence of more than one PRVs, one for every protocol to be supported by the authentication policy. The PRV is the only component that directly or indirectly works with the authentication factor.

The subsequent phases are driven by the authentication policy with the objective to establish an understood level of confidence on an ability. It should be noted that authentication procedure are protocols, so it could be possible to apply an ability based system to the establishing an understood level of confidence that a step in a protocol has been executed.

We assume an execution flow similar to that depicted in the figure below, where a PEP leverages on the functionalities offered by an AVW to let the users demonstrate their abilities.



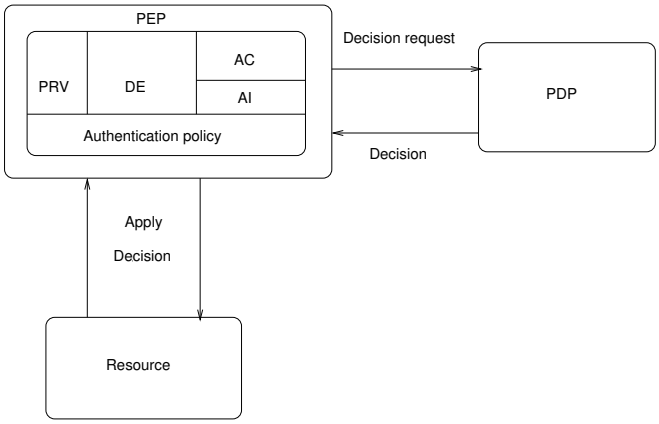
The redirection performed by the PEP are routed to the appropriate PRV, but is is possible to distinguish two types of switching to be performed:

- Mandatory ability: the PEP according to its access control policy redirects to a PRV in order to check a particular ability, that permits to fulfill the

request submitted. Without the corresponding assertion the request could not be authorized, but it could be possible to begin a trust negotiation phase in order to determine, if it exists, an access mode acceptable for the abilities of the user.

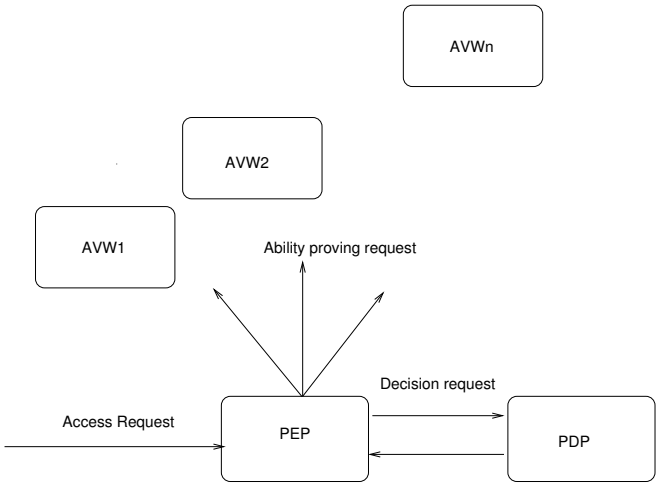
- Negotiated ability: the PEP redirects the claimant to the AVW, but without indicating which ability is necessary to be demonstrated. AVW and claimant perform an authentication protocol, for the release of an assertion. This assertion is presented to the PEP, and subsequently forwarded to the PDP, to determine which is the access right to be granted according to the strength of the ability demonstrated. This case cover the needs of mobile claimants, who are unable, or do not want to use a particular authentication protocol, for example they have not a reader for their smart card, but need to access to the information.

The previous figure is directly inspired to the idea of separation between authentication and authorization, nevertheless it could be possible to consider a common example borrowed from nowadays, where authentication is part of the PEP functions, not directly separated from it:



The assertion issued by the AVW could be used to create a federation of AVWs to build a federation based on the abilities. In this case, the AVW that authenticate the user does not belong to the security domain of resource, but to a third trusted one. This trust decision has to be explicit stated in a local trust policy, which establishes which AVW is considered authoritative to assert a particular ability. In other words the task performed by the AVW are similar to those performed by the identity providers in a federated identity model, but they are intended to transfer capabilities to perform protocols between security domains. In this model claimants before making requests for resources, habilitate to AVW located far from the resource, for example, it could be located in the administrative domain of the claimant; while requesting for access users presents their already obtained assertions. In this way, the access control problems becomes quite similar to a trust management one, because the focus is to

determine if an ability issued by a stranger AVW is to be considered trusted to perform a particular request. This situation is briefly illustrated in the figure below:



3 Other application scenarios

Although ability is a term not present in the literature, we present how AVWs tasks could be used to model some concepts already presented in the area of authentication. The two examples presented below are to be considered a coarse grained application of authentication as protocol role checking.

3.1 Credential Service Providers as AVW

In order to be able to use an authentication factor, a claimant had already performed an other protocol with a different entity than the verifier; in [2] this new entity is denoted as issuer, while the protocol is indicated as registration. Even if the registration could not be entirely automated, it could be modeled like an ability, in fact what the issuer check is for the ability to perform the protocol for the release of the authentication factor. Through the registration, an entity, a Credential Service Provider (CSP), assign to the user an authentication factor, or better, registration is a process performed by the user with a trusted identity, the Registration Authority (RA), that works on behalf of the CSP, with the intent to assign an authentication factor, to be used in a subsequent authentication protocol. Once the RA has attested this ability, the CSP can issue the authentication factor, authorizing the user to become a claimant for it. According to the terms introduced above, the RA is an Protocol Role Verifier and the CSP is the Assertion Issuer.

Consider for example a public key digital certificate, this credential is issued by a Certification Authority to attest the ability of an entity to bind itself with a key pair; to show this ability, the user had executed the enrollment procedure as stated in certificate practice statement of the PKI, so it has demonstrated to be able to perform a protocol, for example he had showed to be able to sign with the corresponding private key the public key to be certified, as in PKCS#10 standard. To witness this ability, the CA issues the certificate, that the claimant is now able to use in other authentication protocols.

To recap, the RA (the PRV) check for the ability to perform the enrollment (a protocol), and once established it, a CA (the AI) issues a certificate (the assertion) to witness the ability.

3.2 Modeling the NIST authentication model with authentication levels

National Institute of Standards and Technology (NIST) draft [2] on authentication presents the model to be used by the government agencies in order to perform electronic authentication; the focus is to remote authenticate users (identity authentication) for electronic government purposes, while other types of authentication like machine to machine are not covered here. The document use an already present classification in the area of federal government [3], classifying the authentication mechanisms in four categories according to the degree of certainty in establishing the identity of the individual and in determining that the individual who uses the credential is the same to whom it was issued.

Authentication policy of an ability oriented system could be considered an alternative way to define these four levels, in fact, it registers the choices ranked by strength of the procedure to obtain a given assertion. In accordance to this, we imagine to group these preferences by using a syntactic construct provided by the language used to express the policy, with the intent to have more than one protocol in each of the four level. A level is determined by the assertion to be issued, so obtaining an assertion from an AVW corresponds to establishing with sufficient degree of certainty the identity of the individual and the possession and control of the authentication factor used. The protocols should therefore be categorized as:

- Level 1: Protocols that assign an identity without the establishment of a link with the authenticated individual. A typical example of these could be username password based authentication protocol, that require only to check the correspondence between what is asserted and what is stored in an archive.
- Level 2: Protocols that require to verify and check the identity information of the user that the authentication factor is destined to, but does not require to release a cryptographic based one.
- Level 3: Protocols that release an assertion after having performed a cryptographic based protocol. In this case, by demonstrating the ability to

perform one of these protocols, it is deduced that the key is owned and in control of the claimant.

- Level 4: Protocols that release an assertion after having performed a cryptographic based protocol with the support of appliance validated according to some specific guidelines, like FIPS 140-2.

4 Related Work

Authentication Authorization Accounting (AAA) systems for network accessible services are presented in [5], [6], the focus here is to provide a common service of AAA for multiple applications. It is assumed that users are authenticated, but how this process should be realized is not deeply investigated, but it is mentioned that could be more useful to authenticate not only the identity of a user, but other types of information too. The cited documents introduce three execution flows denoted as "agent" , "pull" and "push"; the latter two in particular could be considered instances of the architectures presented in the paragraph above, because both are based on the idea of obtaining a proof of the already done authentication. We do not investigate on the requirements of other type of environment presented in those RFC, like those for roaming, but we think it could be possible to match them by placing in an "right" position, with the "right" trust relationships, a proper AVW.

An approach to context authorization is presented in [13], which introduces an access control system inspired to Role Based Access Control (RBAC) for web-based collaboration environment. Here the term context is related to the procedure used to authenticate, but without introducing the concept of an authentication policy. Supporting multiple authentication procedures permits to refine some basilar concept of RBAC, one of the main contributes of the cited work is authentication as automated role activation. A user activates as a role member through the use of an authentication protocol, in this way, authentication maps exactly to role activation, which is a useful solution to enforce the principle of least principle. After activating as a role member the user gains the access rights that has been assigned to that role: in other words, access rights depends on the authentication procedure performed (the context). Note however that if the user is not able to perform a particular procedure, he/she is not able to activate that role, so he/she can not use the rights of that role.

An ability based system can be used to model this type of systems, but with adding the possibility to map a user into a role by means of different procedures. This possibility, not a mandatory feature, can be obtained by the use of an authentication policy, which could lead the role activation phase; it is assumed that using this feature must be preceded by a knowledgeable security management approach.

Trust management systems [7], [8], establish a common framework to authorize operations submitted by credentials. These solutions merge authentication and authorization, by using a policy, as the source of authority according to which the PDP establishes its decisions. Policies and credentials are special type of assertion, which permits to authorize actions requested by principals, the main difference between the two is that policy are locally trusted assertion while credentials are signed assertion that permits to delegate trust. It is possible to observe some similarities between trust management systems and ability systems: although abilities are generated from an authentication process, means that we distinguish in a clear way authentication and authorization, the assertions generated by the AVWs are to be authorized by a PDP according to some access control policy. According to this, it is possible to formulate a trust management like question on the authorization of the abilities like "is set of abilities A associated with a request R compliance with a local policy P ? ": in other words, we feel quite confidence to use a trust management system to be able to authorize abilities, in conformance with the concept of "proof of compliance" [7]. According to this view, abilities, or better, assertions issued by AVW, could be more useful to authorize actions requested by principals in respect of a cryptographic key; note in fact that the task performed by the AVW is not only policy driven, but quite similar to that performed by a trust management engine. Before issuing an assertion the AVW checks if a correct authentication protocol instance exists and in this case it follows the authentication policy to establish which ability has to be certified, in other words if the program executed with the associated assertions (the authentication factors provided) comply with local authentication policy. Note however that TM systems are focused on authorizing rather than authenticating, and considering the AVW like an application that calls a TM engine to authenticate could introduce a considerable overhead.

References

[1] S. Kent, L. Millet Editors *"Who goes there ? Authentication through the lens of privacy"* Nation Academic Press, 2003.

[2] Nist Draft *Special Publication 800-63, Recommendation for electronic authentication* NIST, 2004.

[3] OMB M-04-04 *E-Authentication Guidance,* December 2003.

[4] R. Yavatkar, D. Pendarakis *A Framework for Policy-based Admission Control*, January 2000.

[5] C. de Laat, G. Gross, L. Gommans, J. Vollbrecht, D. Spence Generic *AAA Architecture,* August 2000.

[6] J. Vollbrecht P. Calhoun S. Farrell, L. Gommans, G. Gross, B. de Bruijn, B.V. C. de Laat, M. Holdrege, D. Spence *RFC 2904 AAA Authorization framework,* August 2000.

[7] M. Blaze, J. Feigenbaum, J. Ioannidis, and A. Keromytis *The Role of Trust Management in Distributed Systems* Security Chapter in Secure Internet Programming: Security Issues for Mobile and Distributed Objects, (Vitek and Jensen, eds.) Springer-Verlag, 1999.

[8] N. Li and J.C. Mitchell, *RT: A Role-based Trust-management Framework* DARPA Information Survivability Conference and Exposition (DISCEX III), April 2003.

[9] B. Pfitzmann M. Waidner, *Federated identity-management protocols ? Where user authentication protocols may go.* In 11th Cambridge International Workshop on Security Protocols, Cambridge (UK), April 2003, Lecture Notes in Computer Science. Springer-Verlag, Berlin Germany, 2004.

[10] N. Li, *Delegation Logic: A Logic-Based Approach to Distributed Authorization* Ph.D. thesis, New York University, September 2000.

[11] D. Pinkas, R. Housley *RFC 3379 Delegated Path Validation and Delegated Path Discovery* September 2002.

[12] J. Feigenbaum *Towards an Infrastructure for Authorization,* 1998 USENIX eCommerce conference.

[13] R.Wolf, M. Schneider *Context-Dependent Access Control for Web-Based Collaboration Environments with Role-based Approach* Second International Workshop on Mathematical Methods, Models, and Architectures for Computer Network Security, MMM-ACNS 2003.

[14] M. Kudo, S. Hada, *Access Control Model with Provisional Actions* IEICE Trans. Fundamentals, Vol. E84-A, No. 1, 2001.

[15] J. Lewis *Enterprise Identity management: it's about the business* Burton group research and security strategies, July 2003.

[16] Liberty Alliance I*D-FF Architecture Overview, Version 1.2* http://www.projectliberty.org/specs/liberty-idff-arch-overview-v1.2.pdf

[17] Liberty Authentication *Context Specification, Version 1.2* http://www.projectliberty.org/specs/liberty-authentication-context-v1.2.pdf

[18] *Microsoft .NET Passport Review Guide* http://download.microsoft.com/download/a/f/4/af49b391-086e-4aa2-a84b-ef6d916b2f08/passport_reviewguide.doc

[19] M. Erdos, S. Cantor *Shibboleth Architecture v5* http://shibboleth.internet2.edu/docs/draft-internet2-shibboleth-arch-v05.pdf