150

150

Authorization with Security Attributes and Privilege Delegation

–Access Control beyond the ACL –

Yoshiki Sameshima

Research & Development Department, Hitachi Software Engineering Co., Ltd.
6-81, Onoe-cho, Naka-ku, Yokohama, 231, Japan

Peter Kirstein

Department of Computer Science, University College London
Gower Street, London, WC1E 6BT, England

Abstract

This paper focuses on authorization in distributed environment; the typical authorization scheme employs access control lists, however, the scheme has problems when it is applied to a large scale network. The authors introduce a new authorization scheme, compare it with the old scheme, and present an implementation of an information server which adopts the new scheme.

As a part of authorization, delegation of privileges is important, however, current delegation mechanisms have problems when the delegation crosses a boundary of security domains. The authors propose a solution which refers to security information of other security domains through a directory service.

Keywords: **authorization, privilege, delegation, privilege attribute certificate, access control decision function**

INTRODUCTION

While problems of authentication across computer networks have been received much interest in recent years and there are several standards such as the

Kerberos Network Authentication Service [1], the Distributed Authentication Security Service (DASS) [2] or Open Software Foundation's Distributed Computing Environment (OSF/DCE) [3], distributed authorization (access control decision) is still under discussion in the standardization bodies [4] [5].

There are significant works on authorization in a centralized system; for example, Boolean Expression Evaluation (BEE) [6] [1] introduced generalized policy free access control mechanism, and a unified solution of Mandatory Access Control (MAC) and Discretionary Access Control (DAC) policies is proposed in [7]. However, they cannot be extended to a distribution system environment in a straightforward way because delegation of privilege is out of scope and the representation of authorization information is simple and does not have enough ability to express the semantics of privileges of different security domains.

The most typical authorization scheme makes use of the Access Control List (ACL), which is adopted in the OSF/DCE security architecture [3]. The ACL scheme, which makes the authorization based on user's identity or group, fits an environment where the number of users is relatively small, such as a local area network in an office. However, in the environment of an organizational scale network to which thousands of hosts are connected or a much bigger scale network such as the Internet, the ACL scheme may not be appropriate because the authorization may be required to depend on not only a user's identity but also on various information such as a user's role, a network location, a privilege class, an access time.

The OSF/DCE security architecture and the Secure European System for Application in a Multivendor Environment (SESAME) [8] have adopted the Privilege Attribute Certificate (PAC), which contains user's privileges, restrictions on the privileges and identifiers for auditing and charging, is well structured to transmit authorization information of the user. However, the architectures do not specify the access control information of objects being accessed such as files, application entities, nor how the authorization is made.

The authors propose a combination of the PAC and the Control Attribute Package (CAP) [18] to realize authorization on information servers in an organizational scale network and a worldwide network, and show how the combination fits such information servers. The authors also introduce a document server which makes use of the combination.

Delegation of privilege is an important element of security of a distributed system; the delegation happens when an entity asks another entity to work for

[1] The authors admire anonymous reviews of the Internet Society who pointed out the reference.

the first entity. The typical example is a printing service; a printer server being asked to print a user's file accesses the requested file on a file server which the server does not normally have the access permission.

Current delegation mechanisms adopts chained PACs or proxies which contain privileges and restrictions on the privileges and are send to a target server [8] [9] [10]. The server must verify the chained PACs or proxies and decides whether an accessing entity, for example the printer server, has access permission or not. However, the verification may fail when the accessing entity and the server belong to different security domains, because the protection mechanisms of PACs or proxies, the privileges and the restrictions may be different in each domain.

The authors propose a new delegation mechanism which uses a directory service to retrieve security information of other domains and translates the chained PACs or proxies to a single one which the server can verify and check the permission effectively.

First section illustrates authorization requirements for information servers in an organizational network and in a large scale network, and points out their features. Next section describes deficiencies of the ACL scheme and proposes the PAC/CAP scheme which is proper for the requirements. After problems of the current delegation mechanisms as well as a solution are given, an implementation of an authorization function and a document server which uses the function are described.

AUTHORIZATION REQUIREMENTS FOR INFORMATION SERVERS

In the recent few years, a number of information systems such as World Wide Web (WWW) [11], Gopher [12], Wide Area Information Server (WAIS) [13], have deployed widely and various information is stored and retrieved over the Internet. However, there is a little consideration of security on the access; examples of the consideration are the authentication, integrity check and confidentiality services on the Hypertext Transfer Protocol (HTTP) [14] and the Secure Socket Layer (SSL) over reliable transport protocol [15].

With regard to authorization, there is less consideration; authorization is a process to grant an access of a subject, such as a human user or a client entity, to an object, such as a file or a server entity. For example, many operating systems realize authorization of file access based on the Access Control List (ACL); an ACL entry, which is attached to an object, consists of a subject identity and permitted types of operation to the object, and the operating system decides whether a subject is allowed or rejected to access the object by referring to the ACL. However, the application of the ACL scheme to the information servers does not seem to work, because for the information servers in an enterprise network or even bigger network where there are thousands or millions hosts and users, it is impossible to make access control decision depending on a user's identity or a group to which the user belongs. The following two sections illustrate authorization requirements for such information servers.

Requirements for Information Server in Organizational Network

The followings are requirements for a server running on an organizational network which may consist of thousands hosts and provide information to thousands users in the organization, such as document servers, personnel information servers.

- The server controls a user's access according to categories of the information and need-to-know of the user; for example, people in a development department may not be permitted to access to accounting information because their need-to-know is different from the category of the information.
- The server grants a user's access by comparing the clearness of the information and the clearance of the user; for example, very few people can access "top secret" information.
- The server restricts a user's access depending on a user's identity, a role, a group, an authentication level, a network location etc.
- The server may permit a delegated access from an entity which works for a user who has the access right; for example, a printing entity may be enabled to read a file of which the requesting user has the read permission.
- The organizational network may be divided into a number of security domains, where the categories of information, the role names may be different.

Requirements for Information Server in Large Network

The followings are requirements for a (commercial) information server which runs on a large network and provides information to a large number of users. Examples of such a server are WWW, Gopher and WAIS servers in the Internet which store various information, such as on organizations, people, research or commercial activities, press releases, products, documents.

- A user may pay a fee and get read permission to a set of information on the server.
- The server may distinguish users with their privilege classes which may be different depending on the fee; for example, a privilege class "A" user paying a large fee may be allowed to access a larger set of information than a privilege class "B" user.
- Only a small number of managers have write or modify permission.
- Each user's privilege has an expiry date.
- The number of information units may be very large but the information may be divided into less number of classes from the viewpoint of authorization management.
- A user whose payment is small may be restricted on the access ability or availability (ex. available when the server's load is low).
- A user may access the server from other security domains, often with security policies different from those in the server's domain. As a result, the user's client and the server may support different sets of encryption algorithms and trust different authorities.

Features of Authorization of Information Servers

While in the ACL scheme the access is controlled with the identity of the user or the group to which the user belongs and the operation type, the information server, or more precisely the accesses control decision function (ACDF) [5], is required to authorize the access depending on various attributes of the user (subject), the information (object) and context such as an access time or a subject's location. As a consequence, it is necessary to represent varying characteristics of the subject and the object, and to decide the access control according to the characteristics.

PAC/CAP AUTHORIZATION SCHEME

Mismatch of PAC and ACL

The OSF/DCE security architecture [3] and the SESAME architecture [16] have adopted the Privilege Attribute Certificate (PAC) to specify and exchange the subject's characteristics; a unit of the characteristics is called a privilege and represented in the form of a security attribute, which consists of its type, value and additionally an authority which provides the semantics of the attribute. A set of privileges with control information is signed by an authority of the security domain to which the subject belongs [17]. The signed PAC is distributed to the subject and the subject presents the PAC to the application server. The server verifies the PAC and grants the subject's access depending on the privilege information and the access control information of the object. The PAC has the following properties:

- The PAC provides privileges of the subject, such as a role of the subject, groups to which the subject belongs, need-to-know of the subject.

- The PAC may specify conditions which the subject should satisfy, such as the minimal authentication level, network location (access point).

- The PAC may restrict characteristics of the object for which the PAC is valid, such as an object's name, a service type which the server supports.

- The PAC may specify a time interval in which the PAC is valid.

- The PAC contains not only privileges of the subject but also restrictions on the use of the privileges; restrictions may be added when the privileges are delegated from the original subject to another entity which works on the behalf of the subject. A typical example of the restriction is an access type such as "read only". The next section discusses the delegation of the privileges.

- The PAC is signed by an authority to prevent unauthorised modification by the subject or during transmission and to make sure such PAC is valid.

While the privilege attributes can represent the characteristics of the subject, there is no appropriate representation of characteristics of the object. The OSF/DCE architecture has adopted ACLs that specify which subject, group or role is permitted to access the object with types of operation. However, authorization based on context information such as time, subject's location or a combination of such information cannot be represented with the ACLs. Moreover the ACL does not directly match the privilege because the privilege represents the subject's right of access, while the ACL represents permitted operations of the subject.

Combination of PAC and CAP

The Control Attribute Packages (CAPs) can solve the mismatch between the privilege attributes and the ACLs. A CAP, which is attached to an object, is a sequence of security attributes which represents object's characteristics, or characteristics of subjects which are permitted to access the object.

Category=Accounting, Role=Manager, SubjectLocation=LocalNetwork;
Category=Accounting, Role=Manager, AuthenticationLevel=Strong

Figure 1: Example of CAP

Figure 1 is an example of CAPs which requires that the subject's need-to-know should include "accounting" and the subject's role be "manager" and its location in "local network" or strongly authenticated. While the role information can be handled in the ACL as a subject identifier, the category and the context information, such as subject location, authentication level, cannot be represented with the ACL.

The original document of the CAP [18] specifies the data type of the CAP but does not specify how to compare the security attributes in the CAP and the PAC, nor how to grant access according to them. The authors have refined classes and semantics of the security attributes in the PAC and the CAP, and specified how to make the access control decision; the security attributes have been divided into the following six classes:

- **privilege**: A subject's privilege is represented with a privilege security attribute and transmitted in the PAC. The privilege represents access right or capability of the subject; examples of the privileges are: subject's clearance, need-to-know, role, group to which the subject belongs, etc.
- **positive restriction**: A restriction is represented with a positive or negative restriction attribute and transmitted with the privileges in the PAC. A positive restriction attribute describes a restriction in positive form, that is, represents a restricted privilege. Examples of positive

restrictions are: permitted access time, subject's location, validity time of the PAC, subject's name, target name or type (ex. file name, server service type), etc.

- **negative restriction**: A negative restriction attribute expresses disallowed privilege including: prohibited access type, time or access point, etc.

- **condition**: Security attributes in the CAP are divided into two classes: condition and exception. A condition security attribute of an object characterizes the object, a subject which access to the object is permitted, or status which must be satisfied. Examples are clearness, categories, required minimal authentication level of the subject, access time, etc.

- **exception**: An exception security attribute of an object is a negative form of the condition; it describes a characteristic of a subject which is prevented from accessing the object, or prohibited context status including: prohibited access time, disallowed access point (subject's location), etc.

- **context**: Context information is represented with context security attributes, which are maintained by the server or the ACDF, including: authenticated subject name, authentication level, access count, seal algorithm of the PAC, authority name that issued the PAC, charging identity included in the PAC, access type, arguments of the operation, server's load, etc.

The ACDF is called with six arguments, namely privilege, positive restriction and negative restriction in the PAC, condition and exception attached to the object being accessed by the subject and context information. Each of the condition and exception security attributes is compared against an attribute in the privilege or context class, and each of the restriction security attributes against one of the condition or context class, and the access is allowed when the all comparisons succeed.

An attribute may have an ordered value, which matches a higher or lower value of another attribute. A typical example of the ordered value attributes is the combination of the clearance privilege and the clearness condition; both attributes take one of values of "unmarked", "unclassified", "restricted", "confidential", "secret" or "top secret" (in ascending order), and a read access is permitted when the subject's clearance privilege is equal to or higher than the object's clearness condition. Another attribute has a time or a time interval value; for example, a value of the PAC validity time of the positive restriction class is a time interval and compared with the current time of the context class, and the comparison succeeds when the time interval contains the current time.

The rule of the comparisons, that is, which attribute is compared with which attribute and its method is described in tables; description and its semantics are given in a later section.

The combination of the PAC and the CAP has the following advantages:

- **simple semantics**: The semantics of the combination is simple and easy to understand; it is necessary only to compare attributes of the condition, exception, positive and negative restriction classes with corresponding attributes.

- **easy management**: A security domain may require non-standard security attributes and the manager of the domain needs to configure authorization rules. With the PAC/CAP scheme, this is an easy task because the manager only needs to specify matching rules of the security attributes. A configuration example is illustrated in a later section.

- **checking parameters of an operation**: Grant of an operation may depend on the parameter of the operation as well as the operation type. For example, modification of salary to a value exceeding a specific amount may require an extra privilege. This authorization is realized by comparing a parameter attribute of the context class against a limit attribute attached to the object in the condition class.

- **support of various syntax**: An attribute value is not limited to a string or an integer; it can be an arbitrary type. It is also possible to define the ACL type attribute which is compared with the authenticated subject identity and the operation type of the context class.

On the other hand, the new combination has the following disadvantages:

- **efficiency**: The PAC/CAP scheme is more complicated and slower than the ACL scheme. However, each attribute of the restriction, condition and exception classes can be compared in parallel and it is possible to make the decision in a reasonable time.

- **mismatch with underlying operating system**: Many operating systems support the ACL for the authorization of file access; this might cause mismatch with the CAP.

PRIVILEGE DELEGATION ACROSS DOMAIN BOUNDARY

Current Delegation Mechanisms

It is a common requirement in a distributed system for a subject entity to request another entity to act for the subject on its behalf. The subject is called an initiator, the requested entity an intermediate, and the object a target. The most typical example is a printing service; an initiator asks a printing scheduler to print a file, the scheduler allocates the task to one of printer servers, and the assigned printer server reads the requested file from the target file server and prints it. In this case, the second intermediate, the printer server, needs the read permission of the file on the target.

The SESAME architecture [16] has adopted the PAC chaining method; the chained PACs, which represent the delegated privileges, are included in the PAC for the intermediate. This makes the intermediate can use the initiator's privileges with any action and makes it possible to trace the delegation route which is required for auditing and charging.

The initiator may want to make sure that the intermediate cannot use the privileges for other purposes than the requested action. This is accomplished by specifying restrictions on the privileges. Typical examples are restrictions of access type (read + write → read only) and target (non-restriction → specifying a target or a target service type).

In the Distributed System Security Architecture (DSSA) [9], an initiator generates and signs a certificate to allow the intermediate to act on the initiator's behalf. Restrictions on time are included in the certificates, however, ones on targets or access rights are not well formalized.

original proxy: $[privilege, K_{initiator}]_{K_{authority}}$
delegated proxy 1: $[restriction1, K_{intermediate1}]_{K_{initiator}}$
delegated proxy 2: $[restriction2, K_{intermediate2}]_{K_{intermediate1}}$

$[I]_K$ stands for information I sealed with key K.

Figure 2: Chained Proxies

A similar method, which is based on proxy, is proposed in [10]. A proxy is a certificate that allows the intermediate which has the proxy key to operate with privileges of the initiator which granted the proxy. The proxy is protected from unauthorised modification by adding a seal of the grantor.

Figure 2 illustrates chained proxies, which implement a delegation from *initiator* to *intermediate*2; the top proxy specifies that the authority of the security domain permits *initiator privilege*, the next proxy signed by *initiator* specifies that *initiator* allows *intermediate*1 to use *privilege* with *restriction*1, and finally the last proxy designates that *intermediate*1 grants *intermediate*2 to use the privilege with *restriction*2. All three proxies are sent to the target, which verifies the proxies and checks whether *intermediate*2 has or not *privilege* with *restriction*1 and *restriction*2.

Deficiencies of the Current Delegation Mechanism

The above delegation mechanisms have the following deficiencies in the case that the authority, the initiator, the intermediates and the target do not belong to a same security domain:

- **policy, authority**: While PACs are issued from a Privilege Attribute Server (PA-Server) in the SESAME architecture, proxies are generated by the initiator, the intermediates or an authorization server in the proxy-based authorization scheme. However, the target domain does not accept the PACs or the proxies because the security policies of the domains may be different. Moreover the verification of seals generated by principals or authorities of other security domains may fail since the target may not trust authorities in other domains.

- **seal algorithm**: The seal algorithm of the PAC may be different in each security domain and the target may be unable to verify the seal and thus to check the chained PACs or proxies.

- **privilege and restriction mapping**: Each security domain may define privileges and restrictions of its own. Again the target may fail the translation of privileges or restrictions of the different security domain to local ones.

- **complicated computation of restrictions**: The checking of the chained privileges and restrictions may be complicated and cost much. For example, in Figure 4 the original privilege, *privilege*, does not have a restriction on the target, *restriction*1 added by *initiator* limits the target to printing service entities in domain A and the final target (file server1), and *restriction*2 added by *intermediate*1 (printer scheduler) limits the target only to file server1. The final target in domain B, *target*, needs to verify the fact that *intermediate*1 provides a printing service in domain A. However, the fact is about the different domain A and *target* in domain B needs extra information to verify the fact.

In order to solve these problems, the SESAME architecture provides an inter-domain server which can verify seals of PA-Servers of other domains and if necessary translate privileges and restrictions to the local representation. For the purpose of these services, the inter-domain server must support the seal algorithm of PACs generated by the PA-Servers, get the PA-Servers' keys, know the mapping information of privileges and restrictions between the local domain and other domains. However, this is not always true in a large network environment which consists of many different security domains.

A New Delegation Mechanism

A more practical solution is that the chained PACs or proxies are translated to a single PAC or proxy which can be verified by the target, and the translation is done in the initiator's or the intermediate's domain. If necessary, the keys or the certificates of the keys issued by a trusted third party may be attached to the PAC [2].

The refine Privilege Attribute Certificate (refinePAC) service provided by the PA-Server [17] may be used for the translation from the chained PACs to the single PAC; this service is originally intended to tailor set of privileges and controls the PAC, such as longer validity time, depending on the applications which the initiator wants to use. However, with the information of policies, trusted authorities, supported seal algorithms, privileges and restrictions of the final target, the refinePAC service can generate a single PAC which can be verified by the target, and can solve the problems mentioned in the previous section.

The key point is the information of the policies, the trusted authorities, the seal algorithms, the privileges and the restrictions which is used for the translation. In the case that the two domains trust each other, the information is held locally and the refinePAC service is enforced to use the seal algorithm, the privileges and the restrictions supported by the target domain. When both domains do not trust each other, the refinePAC service needs to find a trusted third party which is trusted by the two domains and needs to know

[2] While a certificate of a public key is well known and defined in X.509 standard [19], a discussion of a secret key certificate can be found in [20].

seal algorithms, privileges and restrictions which are commonly supported by both domains. The information can be retrieved via a directory service; each security domain needs to announce the following information:

- **policy**: The security policy identifiers which specify policies supported by the domain with qualifier information pertaining to the policies; the syntax of the information is given in the amendment to X.509 [21]. A security policy may specify acceptable use, cryptographic algorithms, user certification procedures or operational matters such as validity time length of certificates, etc. A domain may also specify prohibited policies which the domain cannot accept. The PA-Server needs to check that there is no contradiction between its policies and those of the target domains.

- **authority**: A set of authorities trusted by the domain; the authority may specify the security policies which should be supported by domains trusting the authority. The PA-Server of the initiator or the intermediate needs to find an authority in the set which is also trusted by the domain of the target.

- **seal algorithms**: A set of seal algorithms which the domain supports; the PA-Server uses an algorithm in the set which the server supports. In most case the algorithms supported by the commonly trusted authority will be used. A security policy of a domain may specify the precedence of algorithms.

- **privilege and restriction**: A set of privileges and restrictions which the domain supports and optionally mapping information between standard ones and ones defined locally; the PA-Server translates local privileges and restrictions to the remote ones. In many cases, the privileges and the restrictions defined by the commonly trusted authority or supported policies will be used.

Note that the above information may need to be protected; with forged information, the refinePAC service generates a PAC which cannot be accepted in the target domain, and this leads to denial of services.

This method has the following advantage and disadvantage:

- **verification cost**: The verification cost of privileges and restrictions at the target side is reduced because the chained PACs and proxies have already been deleted at the initiator's side and the target can check authorization immediately after the verification of the single PAC. Instead of this, the cost of issuing the single PAC increases at the initiator's side. However, normally this is not a problem because requests coming from many initiators make the target system's load heavier than the initiator's in general and it is desirable to distribute and reduce the cost of the verification. This is also true about the cost of the mapping of the privileges and the restrictions.

- **trace information**: The translation of the chained PACs to a single PAC makes the trace of the delegation path impossible. However, adding a new parameter, which includes the initiator and the intermediate(s), filled by the PA-Server to the PAC makes possible to trace the necessary information for auditing and charging.

The ECMA standard [18] has proposed another inter-domain service mechanism; the initiator inter-domain service translates local privileges to standard ones and signs the PAC with the inter-domain service key, next the PAC is passed to an inter-domain server of a trusted third party. The PAC is verified and signed by the authority of the trusted third party, and send back to the initiator through the inter-domain server. Finally, the re-signed PAC is passed from the initiator to the target and the inter-domain server in the target domain, and is verified and translated to local privileges and restrictions.

The differences between the ECMA's mechanism and the authors' are the route of PACs and certificates included in the PAC. While in the ECMA mechanism a PAC is routed from the inter-domain server, the trusted third party, the inter-domain server, the initiator and finally to the target domain, in the new mechanism the PAC is directly passed to the target. The seal generated by the PA-Server in the initiator domain can be verified by the target because the PA-Server adds the key certificate of the server issued by the authority of the trusted third domain to the PAC. The privileges and restrictions contained in the PAC are acceptable in the target domain, however, the target or the inter-domain server of the target may translate them to local ones.

IMPLEMENTATION

The PAC/CAP authorization schema has been implemented on a WAIS server; a generic ACDF has been implemented and the WAIS protocol (the initialization phase and the document retrieval phase) has been extended to support the new authorization mechanism. Each of them is described in the following sections. As for the new delegation mechanism, the new refinePAC service which translates chained PACs into a single PAC will be implemented later.

Access Control Decision Function

The function takes six sequences of security attributes as arguments, namely, attributes of the privilege, positive and negative restriction, condition, exception and context classes, and returns OK which means the access is allowed, NOTOK indicating the access denied and UNKNOWN when unrecognized attributes are given.

`category:`	`IncludedSETOFPrintableString:`	`category:prv`
`clearness:`	`SmallerINTEGER:`	`clearance:prv`
`accesstype:`	`IncludeSETOFInteger:`	`accesstype:ctx`
`subjectAddress:`	`IncludeIPAddress:`	`address:ctx`
`permittedAccesstime:`	`IncludeTime:`	`accesstime:ctx`
`minimalAuthenticatedLevel:`	`SmallerINTEGER:`	`authenticatedLevel:ctx`

Figure 3: Example of Condition Attribute Table

The authorization rule is managed by security attribute tables; Figure 3 illustrates how a condition attribute class is configured; each line specifies a condition security attribute and consists of three tuples: the attribute name, the attribute syntax which defines value type and the matching rule of the values, and the attribute name of the privilege (indicated with `:prv`) or context (`:ctx`) class which is compared with the condition attribute. For example, the last line specifies that the "minimal authentication level" condition attribute must be smaller than the "authentication level" attribute value of the context. The authorization rules for exception, positive and negative restriction classes are configured in the same manner.

Initialization Phase

For the purpose of simple description of the extend protocol, notations listed in Table 1 are used in the following.

1. **Initialization Request**

The subject's privileges and the restrictions are transmitted in the initialization phase in the form of the PAC as well as authentication information and a secret session key; all information is carried in the idAuthentication parameter of the InitializeRequest of the WAIS protocol.

$C \rightarrow S$: $\{AuthInfo, PAC, KeyPack\}$

where

$$\begin{aligned} AuthInfo &= \{S, random, time\}_{PrvKey(C)} \\ PAC &= \{serialNumber, P, R, id, PS\}_{PrvKey(PS)} \\ KeyPack &= \{SK, S, time, random\}^{PubKey(S)} \end{aligned}$$

The authentication information ($AuthInfo$) is same as the bind-token of the directory access protocol [23]; it includes the intended recipient (S), a random number and the current time sealed with the client's private key ($PrvKey(C)$) of the RSA encryption algorithm [22]. The client's public key certificate [19] might be attached to the authentication information.

The PAC contains a serial number, privileges (P), restrictions (R), an authority name (PS) and identifiers for auditing and charging (id). The PAC is sealed with the PA-Server's private key ($PrvKey(PS)$) and distributed in a off-line manner. The format is different from one defined in [18]; the validity time is omitted because it is included in the positive restriction, the restriction type is changed because of simplicity and direct comparison against condition and context security attributes, and the contained PAC parameter is omitted because the PAC is always packed into a single PAC described in the previous section and the parameter is not used.

A randomly generated session key (SK), the intended recipient (S), time and a random number are encrypted with the server's public key ($PubKey(S)$) and packed in the session key package ($KeyPack$).

2. **Initialization Response**

   The server verifies the authentication information with the client's public key which might be stored in a local cache or retrieved from the public key certificate attached to the authentication information. If the verification succeeds, the server checks whether the included time ($time$) is enough to close to the current time of the server, and the random number ($random$) was not used before. After verification of the PAC, the server checks the positive restrictions of the PAC which might restrict the subject. In this case the server needs to check the authenticated client equal to the subject.

The server decrypts the session key package with the server's private key, confirms that the included name is same as the server's name, and checks the included time and random in the same way as one of the authentication information. The session key is used to realize integrity and confidentiality of documents during transmission from the server to the client.

The server sends back the normal WAIS initialization response in the current version. For the mutual authentication between the server and the client, the server needs to send back the authentication information of itself in the idAuthentication parameter of the InitializationResponse:

$S \rightarrow C$: $\{S, random, time\}_{PrvKey(S)}$

where the random ($random$) equals to the random in the session key package. With the verification of the information, the client can authenticate the server and make sure that the session key package is correctly decrypted and the session key is shared between the two principals.

Document Retrieval Phase

Each time the client requests to retrieve a document, the ACDF is called with arguments of the six classes of the security attribute, namely, the privileges and the restrictions got from the PAC, context information managed by the server, the conditions and the exceptions of the associated access control class; each document is tagged with an access control class identifier and each access control class is associated with a CAP, a sequence of condition and exception security attributes.

For the purpose of confidentiality of the retrieved document, a randomly generated data encryption key (dek) is used for the encipherment of each document, and send with the document after it is encrypted with the session key. For the sake of integrity, a message digest of the document is encrypted with the session key and send with the document. Since only the client and the server share the session key, the client can decrypt the enciphered document and check integrity. These three components is handled as a single document in the protocol.

$S \rightarrow C$: $\{cek\}^{SK}, \{document\}^{cek}, \{MD(document)\}^{SK}$

Currently the DES-CBC algorithm [24] is used for the two services and the MD5 message digest algorithm [25] in order to generate message digests of documents.

In view of security, the retrieval request should be authenticated; the request should be sealed with the shared session key by the client. However, there is no appropriate parameter it the WAIS protocol, the authentication of the request is not implemented.

CONCLUSION

In this paper the authors have enumerated requirements of authorization especially for information servers running on an organizational scale and a large scale network and pointed out the problems of the ACL-based authorization, and the PAC/CAP authorization scheme has been proposed which has several advantages when it applied to such information servers. Next delegation problems have been pointed out and a solution using translation of chained PACs into a single PAC with help a directory service has been proposed. Finally an implementation of a WAIS server and a client have been presented.

Currently the PAC is signed by the authority according to an asymmetric encryption algorithm and distributed in an off-line manner; in later versions an on-line distribution of PAC supported by the new refinePAC service will be introduced.

Auditing is a new frontier of security which the authors have not addressed yet. The current ACDF library records only what subject with what auditing identifier is permitted or rejected to access to which object. This auditing trail might not be enough because the context information is not recorded which cannot be traced after the decision. The authors will examine what kind of information is necessary during authentication, authorization and real processing of requests from the subject, and implement in later versions.

References

[1] **Kohl, J** and **Neuman, B** *The Kerberos Network Authentication Service (V5)* Internet Requests for Comments 1510 (September 1993)

[2] **Kaufman, C** *DASS - Distributed Authentication Security Service* Internet Requests for Comments 1507 (September 1993)

[3] **Fairthorne, S B** *Security Enhancements for DCE 1.1, OSF/DCE SIG Request For Comments 19.0* (December 1992)

[4] **Abrams, M D** and **Joyce, M V** Extending the ISO Access Control Framework for Multiple Policies *Proceedings of the IFIP TC11 Ninth International Conference on Information Security, IFIP/Sec'93* Toronto, Canada (May 1993) pp 343-358

150

[5] **ISO/IEC** *Open Systems Interconnection - Security Frameworks in Open Systems - Part3: Access Control* Draft International Standard, DIS-10181-3 (1994)

[6] **Miller, D** Access Control by Boolean Expression Evaluation *Proceedings of the Fifth Annual Computer Security Applications Conference* (December 1989)

[7] **McCollum, C, at el** Beyond the Pale of MAC and DAC – Defining New Forms of Access Control *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy* Oakland, California, USA (May 1990)

[8] **Kaijser, P, at el** SESAME: The solution to security for open distributed systems *Computer Communications* Vol 17 No 7 (1994) pp 501-518

[9] **Gasser, M** and **McDermott, E** An Architecture for Practical Delegation in a Distribution System *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy* Oakland, California, USA (May 1990)

[10] **Neuman, C** Proxy-Based Authorization and Accounting for Distributed Systems *Proceedings of the 13th International Conference on Distributed Computing Systems* Pittsburgh, Pennsylvania, USA (May 1993) pp 283-291

[11] **Berners-Lee, T, at el** The World-Wide Web Initiative *Proceedings of INET'93* California, USA (August 1993)

[12] **Anklesaria, F, at el** *The Internet Gopher Protocol, a distributed document search and retrieval protocol* Internet Request for Comments 1436 (March 1993)

[13] **Marshall, P** *WAIS: The Wide Area Information Server or Anonymous What???* (June 1992)

[14] **Rescorla, E** and **Schiffman, A** *The Secure HyperText Transfer Protocol* Internet Draft (July 1995)

[15] **Hickman, K** and **Elgamal, T** *The SSL Protocol* Internet Draft (June 1995)

[16] **Pinkas, D, at el** *Secure European System for Applications in a Multi-vendor Environment - an Introduction, Issue 1.2* (September 1993)

[17] **ISO/IEC JTC 1/SC 21** *Authentication and Privilege Attribute Security Application with Related Key Distribution Functions - Part 3: Service Definitions* Working Draft (November 1993)

[18] **European Computer Manufactures Association** *Security in Open Systems - Data Elements and Service Definitions* Standard ECMA-138, Geneva, Switzerland (December 1989)

[19] **ISO** *Information Processing - Open Systems Interconnection - The Directory - Authentication Framework* IS-9594-8 (1988)

[20] **Davis, D** and **Swick, R** Network Security via Private-Key Certificates *Operating Systems Review* Vol 24 No 4 (1990) pp 64-67

[21] **ISO/IEC JTC 1/SC 21/WG 4, ITU-T Q15/7** *Information Technology - Open Systems Interconnection – The Directory: Authentication Framework, AMENDMENT 1: Certificate Extensions* Proposed Draft Amendment 1 to ITU X.509 (December 1994)

[22] **RSA Data Security, Inc.,** *Public-Key Cryptography Standards #1: RSA Encryption Standard* (1993)

[23] **ISO** *Information Processing - Open Systems Interconnection - The Directory - Abstract Service Definition* IS-9594-3 (1988)

[24] **Open Systems Environment Implementors' Workshop** *Stable Implementation Agreements for Open Systems Interconnection Protocols: Part 12 - OS Security* (1994)

[25] **Rivest, R** *The MD5 Message-Digest Algorithm* Internet Request for Comments 1321 (April 1992)

150

Table 1: Notations

notation	description
$\{I\}_k$	information I signed with key k
$\{I\}^k$	information I encrypted with key k
$PubKey(X)$	public key of principal X
$PrvKey(X)$	private key of principal X
C	client
PS	PA-Server
S	WAIS Server
$AuthInfo$	authentication information
$KeyPack$	session key package
SK	session key
P	privileges
R	restrictions
id	audit identifier and charging identifier
dek	data (document) encryption key
$MD(I)$	message digest of information I

biographical note of Yoshiki Sameshima

Yoshiki Sameshima received his BA from University of Kyoto and MA from University of Osaka in Mathematics. He has been working for Hitachi Software Engineering Co., Ltd. since 1986, where he has maintained the network environment and researched computer networks. He stayed at Computer Science Department of University College London during 1992-1994 and studied network security.

biographical note of Peter Kirstein

Peter Kirstein received his BA in Mathematics and Electrical Engineering at Gonville and Caius College, Cambridge, a Ph D in Electrical Engineering from Stanford U., and a D.Sc from London U in the same subject.

Peter is Professor of Computer Communications Systems and Director of Research in the Department of Computer Science, University College London. From 1980-94, he was the first Head of the Department of Computer Science. He joined the University of London in 1967, first as Reader and then Professor at the University of London Institute of Computer Science, before joining

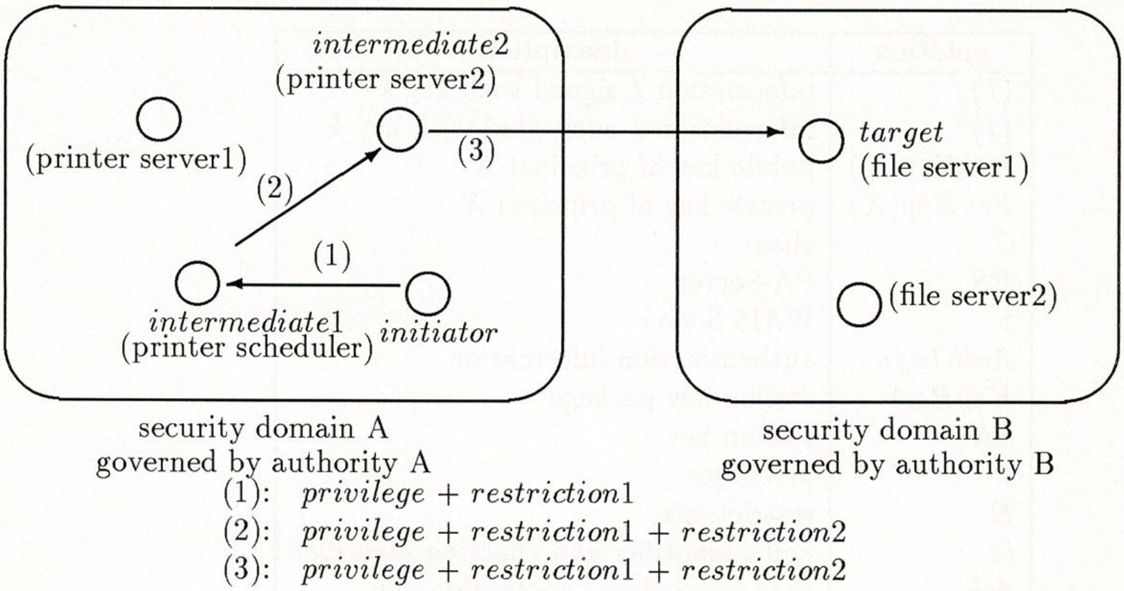

Figure 4: Delegation across Security Boundary

University College London in 1973. Previously Peter has worked at Stanford U (USA) [1957-58], the Centre of European Nuclear Research (CERN) in Geneva (Switzerland) [1959-63], and the US General Electric, located in Zurich (Switzerland) [1963-67].

Professor Kirstein has been leading research projects in computer communications - mostly in collaboration with European and US colleagues. Amongst these activities are developments in multimedia, directory and security applications, and piloting them in the Research Community in Europe and elsewhere. He is Director of the CEC-sponsored MICE project to pilot multi-way, real-time multimedia services in Europe with links to the US. This work arises out of other activities he has been conducting both with DARPA and the CEC RACE program in Network Management and Distributed Real-Time systems. These activities will be extended over the next couple of years over the emerging ATM infrastructures in the UK and the rest of Europe. In the CODA project, he is piloting access to Chemical Journals in ODA. Other recent pilots are the PARADISE project in Directory Services and the PASSWORD project in Security Services.

Peter is a Fellow of the UK Royal Academy of Engineering, the British Computer Society, the Institute of Physics, and the Institution of Electrical

Engineering. He is a Senior Member of the Institution of Electrical and Electronic Engineers.