Security Considerations in a Network Management Environment

When the exchange of management information takes place across the boundary between two separate management domains, and makes use of public data networks, security issues must be considered in greater detail.

Donal O'Mahony

DONAL O'MAHONY is a lecturer in Computer Science at Trinity College.

This article considers the security aspects of communication between two management processes operating in different management domains; identifies two major risks: the security of information exchanged during the management association, and control of access to the management information base (MIB); and enumerates the various threats that must be guarded against and possible methods of attack. Security techniques, including symmetric and public key cryptosystems, are employed in the design of a method of achieving a secure management association. A scheme of authorization control for MIB access is developed based on work carried out on the X.500 directory service.[1]

The management of an Open System's network resources takes place in the context of a Management Association. The resources themselves are controlled by an *agent* process which presents a view of these resources to the outside world as a number of *managed objects*, each of which contains a number of attributes. The collection of objects presented to the outside world by the Agent is known as the *management information base*, or MIB. A *manager* process regulates the operation of the managed resources by engaging in a management association with the agent and instructing it to carry out simple operations such as GET / SET attribute, CREATE / DELETE object, etc., on elements of the MIB. Within a single management domain where all processing nodes and network links are under the control of the same administration, security is not such a critical issue. However, when the management association takes place across the boundary between two separate management domains, and make use of public data networks, security issues must be considered in greater detail.

Scenario Under Consideration

The scenario considered in this article consists of two open systems, operating in separate domains, communicating management information by making use of the Common Management Information Service (CMIS). It is assumed that one of the management systems is operating in a domain that is under the control of a network services provider, telecommunications administration, or PTT. The other management system is operating in a customers management domain.

This article considers the sequence of events that occurs when a fault is discovered at a network element that is a part of the customer's management domain, and where the cause of this fault lies with a network element or service under the control of the administration. In this instance, the customer's management application process (MAP) may form an association with the administration's complaint handling application and report the complaint by performing operations on the remote MIB.

In certain cases, either party may wish to allow the other to selectively access a portion of its MIB. For example, the customer may wish to allow the administration to carry out tests involving equipment at the network access point, or alternatively, the administration may use this facility to advertize planned service outages. A characteristic of this form of communications is that it can take place at any time, and either party can assume the role of manager or agent. From a security perspective, this scenario poses two problems:

- Security of information exchanged during the association.
- Control of access to the MIBs in each domain.

These two problems will be addressed in the following sections.

Threats to be Addressed

The scenario outlined above suffers from most categories of threats faced by generalized distributed systems [1] including the following:

Disclosure of Information — Information held within the management information base of either a customer or administration may well be used by other parties in such a way as to damage their interests or gain competitive advantage over them.

Contamination of Information — This is a complement of information disclosure. Information

[1] *This research is based on work carried out in association with Broadcom Eireann Research Ltd. on the ADVANCE project which is funded under the EC RACE program.*

0890-8044/94/$04.00 © 1994 IEEE

that is valuable to either the customer or administration's organization may become worthless if unauthorized information is mixed with it. Information may also be deliberately contaminated to mislead.

Unauthorized Use of Resources — Access to an organizations MIB may allow an unauthorized user to consume resources or perform actions that would be detrimental to the interests of that organization.

Misuse of Resources — Authorized use of resources may give authorized individuals the opportunity to perform activities that are harmful to the organization. These activities may be intentional or accidental. In the context under consideration, a customer may unwittingly destroy or corrupt all information relating to complaints in progress, or the administration may destroy fault history records held in the customers MIB.

Unauthorized Information Flow — Information flow must be controlled, not just between endusers, but also between endsystems. It may be the case, for example, that information of certain types should not be sent to certain classes of terminal.

Repudiation of Information Flow — This involves denial of transmission. If a change is made to an MIB by an authorized user, it should not be possible for the user later to deny making the change.

Denial of Service — It should be possible to detect any attempt by a third party to deny service to either user of the communications channel.

Methods of Attack

The threats outlined above represent the ways in which the interests of an organization can be compromised by a breach in security policy. We will now describe some of the means by which should breaches could come about.

Unidentified Subjects — It is important that all subjects (i.e., open end-systems and users) be fully identified to the system. This identification must be positive and non-forgable.

Passive Traffic Interception — This covers the case of simple eavesdropping on information sent.

Active Traffic Interception — This involves tampering with the message stream, and could involve insertion of bogus information (e.g., insert a bogus set operation into a CMIS exchange) or the selective removal of information. Simple encryption alone may not guard against this form of attack, as the intruder can easily record part of the exchange, and then replay this at a later time. It is necessary to include sequence information in the encrypted information stream to protect against this form of attack.

Introduction of Unauthorized Resources — If the attacker finds a means of introducing contaminated software or hardware into the management systems of either party, a variety of malpractices can take place.

Traffic Analysis — A simple analysis of the volume and timing of communications traffic emanating from an open system can provide significant clues as to the nature of business being conducted. This form of attack is difficult to guard against, but can be satisfactorily achieved on a link-by-link basis, by ensuring that real data is indistinguishable from idle traffic.

Defense Mechanisms

In the above discussion, I have outlined some of the possible forms of attack that an intruder may employ to compromise the security of a CMIS dialogue. I will now discuss the protection mechanism that should be put in place in order to defend against these.

Peer Entity Authentication — When a CMIS-based management association is to be formed, the initiating systems issues an `M-Initialize.Request` primitive [2]. Associated with this primitive, are a number of parameters including the initiator reference, destination reference, and responder reference. Each of these references can contain a Systems Management Application Entity Title (SMAE-Title) that can be used to uniquely identify that entity. In addition, an access control parameter is also included that can be used to verify that the stated identify is valid. By making use of mechanisms based on public key cryptosystems (outlined later in this article), the source and destination systems can satisfy themselves as to the authenticity of the other party. They can also use the same mechanisms to be able to offer proof of the association taking place, in case the authorized party subsequently denies this.

Data Origin Authentication — In addition to verifying the identity of the requesting application entity, we must also ensure that the transactions are being carried out from the correct location. This guards against such eventualities as: a legitimate user accessing the service at gun-point from a hostile system, or an illegitimate user who has somehow obtained the necessary security parameters to access the system [3]. The various references alluded to above contain presentation service access point (PSAP) addresses that can be used to verify the origin of the association.

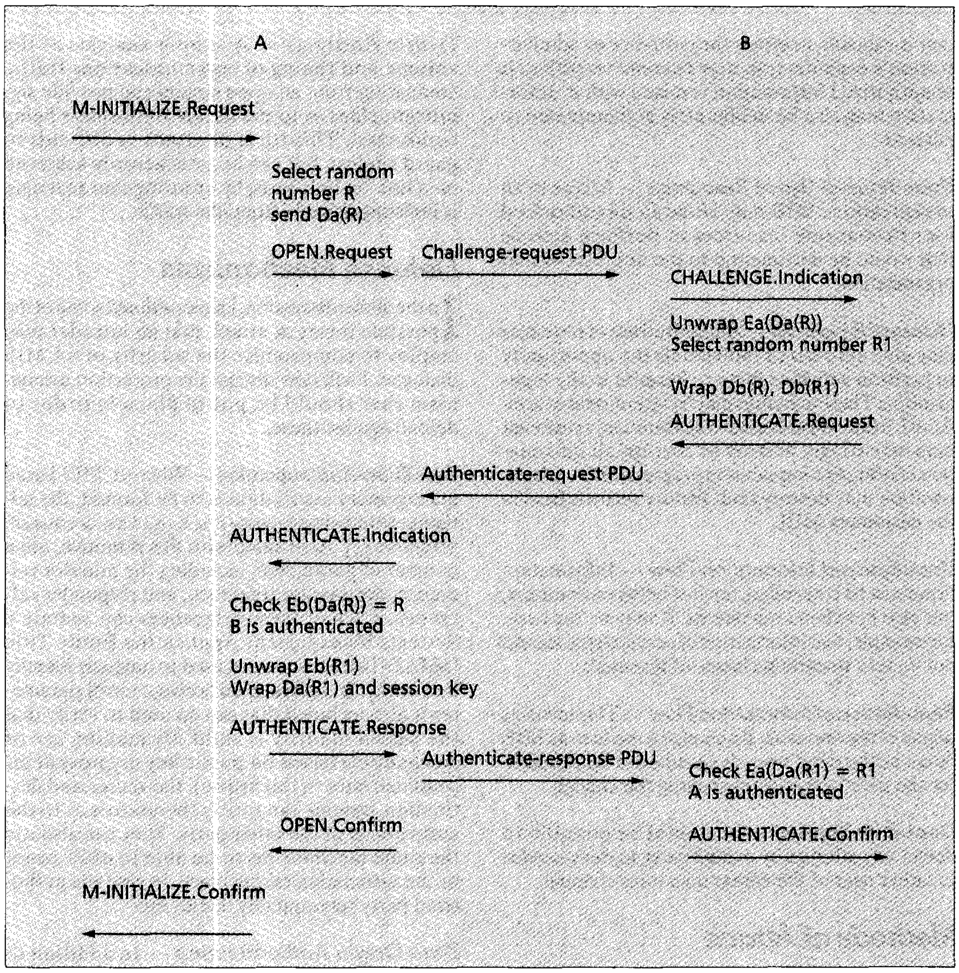
Data Confidentiality — Each of the service requests that form part of CMIS map onto corresponding protocol data units (PDUs) of the Common Management Information Protocol (CMIP) [4]. These PDUs in turn map into those of the remote operations service elements (ROSE). An intruder intercepting a ROSE PDU will be able to gain full knowledge of the semantics of the original CMIS primitive. Accordingly, the content of these PDUs must be enciphered for protection. It should be noted that this encipherment will not protect against attack by traffic analysis, and this will still need to be applied on a link-by-link basis.

Data Integrity Protection — In order to guard against tampering with individual PDUs, or by replaying PDU sequences, the `InvokeId` and access control parameters that accompany each CMIS primitive can be used to ensure that both the content

■ ■ ■ ■ ■

A simple analysis of the volume and timing of communications traffic emanating from an open system can provide significant clues as to the nature of business being conducted.

It is desirable to be able to control access to the MIB, for example, allowing all users to *GET* the value of an attribute, but only a subset to perform the *SET* operation.



■ **Figure 1.** *Management association with peer-entity authentication.*

and sequence integrity of the message stream remains secure. The details of the use of message digest codes will be out- lined later in this article.

Access Control — Having established a secure association between two management processes, the two parties are then enabled to manipulate each others MIBs. It is desirable to be able to control access to the MIB, for example, allowing all users to `GET` the value of an attribute, but only a subset to perform the `SET` operation. The operation of the `M-ACTION` primitives will also need selective access control. These points are explored further later in this article.

Establishing a Secure Management Association

In a conventional association between two management entities, the initiating entity generates an `M-INITIALIZE.Request` primitive specifying the other party. This is mapped to a Remote operations service element `BIND` primitive, which is in turn mapped to an `A-ASSOCIATE.Request` primitive. This causes the destination user to receive an `M-initialize.Indication`, respond with a `M-Initialize.Response`, which culminates with the initiating user receiving an `M-INITIALIZE.Confirm`. This process provides no protection against the types of attack specified earlier in this document. If, for reasons of security, we do not wish to allow associations to take place, unless satisfactory mutual authentication has been carried out, a more complex procedure is needed for association establishment. The following procedure adapted from [5] establishes authenticity using a challenge-response mechanism. If A is setting up an association with B:

- A generates a random number R and encrypts it with his secret key Da. He then sends Da(R) to B.
- B receives this, knows that it came from A, and unwraps it using A's public key.
- B then encrypts this using his secret key Db. He then sends Db(R) back to A.
- When A receives this result, he unwraps it using B's public key and verifies that it is equal to what was sent, i.e., that Eb(Db(R)) = R. If this is true, then A can be sure that B is who he claims to be.

Note that the above procedure must be performed in both directions for two-way authentication to be established.

One possible means of achieving this is shown is Fig. 1 where two parties A and B wish to establish a management association. Each has a public key (Ea and Eb) and a private key (Da and Db). It is assumed that the public keys of any entity can be obtained by requesting a certificate from the X.500 Directory service. First A challenges B by sending it a random number. B responds to this challenge, and generates a new random number that is simultaneously used to challenge A. When A has responded to this second challenge, both parties can be confident that the other is who they purport to be.

One of the later stages in the process illustrated in Fig. 1 involves A sending a session key to B. This would normally be an encryption key associated with one of the symmetric key algorithms such as DES. This can be given to the presentation layer, which will use it to encrypt any subsequent traffic on the association. This will ensure data confidentiality on the association thereafter.

Non-Repudiation of Management Operations

In the context of a secure and confidential management association, one party may invoke management operations (GET, SET, ACTION, etc.) on the other MIB. In the case where either party wishes to guard against a subsequent denial that an operation was invoked, provision must be made to make operations nonrepudiable.

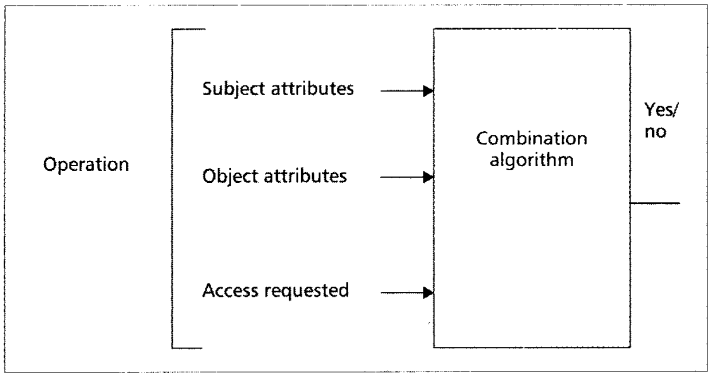
This can be achieved by computing a message authentication code (MAC) for the PDU associated with the management operation. This MAC can then be signed using the secret key of the invoker. The performer can then keep the PDU contents in a log and use it later as proof that the invocation was requested. In acknowledging the PDU, the performer can include the same MAC signed with the private key of the performer, which can later be used as proof of acknowledgment. The only problem to be solved is how to include the signed MAC in the management PDUs. This can be done in two ways:

- Information can be included in either the InvokeID or AccessControl of normal CMIP PDUs. This strategy will work with the Get, Set and Action CMIS primitives, but since the event-report service does not provide this, it will not be possible to provide it with non-repudiation facility.
- There are proposals [3] to extend the capabilities of the OSI presentation layer to allow non-repudiation to be applied to any arbitrary PDU. This is a better solution.

Secure Association Summary

The above discussion has outlined the ways in which two management entities may establish an association with peer-entity authentication. In the process of setting up an association, they can exchange a session key, and use this to ensure data confidentiality and integrity throughout the session. By the application of stream ciphers on a link-by-link basis, they can achieve some protection against attacks by traffic analysis.

The framework outlined above allows two management entities to communicate with a view to manipulating each others' MIBs. Once this is established, the next problem to be solved is how to control and restrict access to the MIB in a selective manner. This problem is dealt with in the following section.



■ **Figure 2.** *Authorization of an operation.*

Authorization Control

In speaking of authorization control, we make an assumption that a subject (e.g., a person or an application program) is requesting access to an object (e.g., a file, a printer, or a managed object), and that the subject has already been authenticated. In this context, three things [1] will be taken into account:

- Subject attributes — for example, the subject name, his role in the system (e.g., customer, manager, fault logging daemon) and possibly his trustworthiness.
- Object attributes — for example, the object's name or its sensitivity (e.g., causes immediate shutdown of all systems).
- Type of access requested — in the context that we are considering, this access will be specified by the type of operation being performed: (GET, SET, CREATE, DELETE, etc.).

As shown in Fig. 2, when an operation is requested, these three items will be combined according to an algorithm dictated by the security policy in force, and will yield a yes or no decision.

All authorization control schemes are based on the above principles. Where they differ is in where the attributes relating to the subject and object are stored. The optimum location for these attributes will depend on the pattern of access (e.g., ratio of subjects to objects) and on relative importance of access criteria (e.g., speed of authorization versus the ease with which the attributes can be updated.

One major goal in incorporating authorization control into the OSI management framework, is that it should be done with the minimum of disruption to existing standards. A similar constraint was encountered by the designers of X.500-based directory systems, and the solution to that problem seems highly applicable in a management context. In the next section we will examine how designers of the X.500 implementation known as QUIPU addressed this problem, and at how this solution can be adapted to provide effective authorization control for a MIB.

QUIPU Security Model

The X.500 Directory Recommendations describe a method of implementing a distributed database containing details of people, organizations, countries, application processes, and other entities of interest from a communications perspective. Its model for representing this information has many similarities to the the way in which management information is represented in an MIB, and indeed the X.500 recommendations exerted a great influence over those drawing up the management standards.

One major goal in incorporating authorisation control into the OSI management framework, is that it should be done with the minimum of disruption to existing standards.

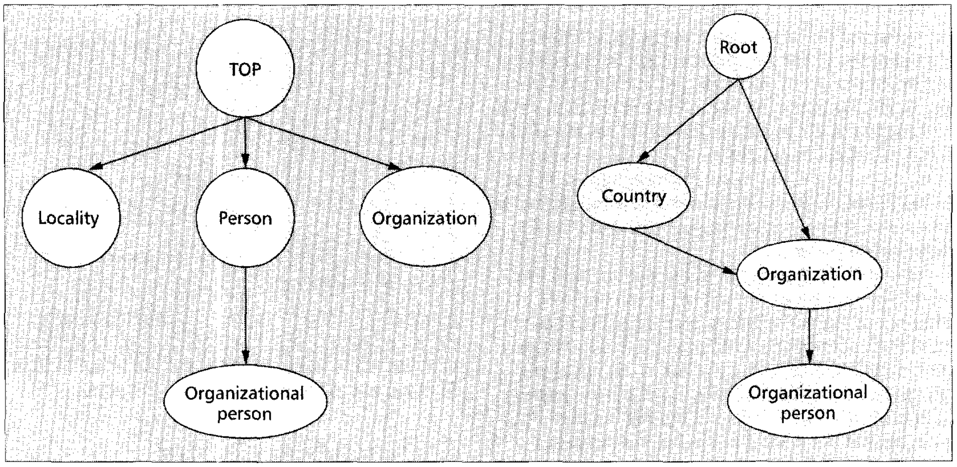


Figure 3. *Portions of the X.500 object class hierarchy and suggested DIT structure.*

Just as in an MIB, the items of interest in a directory are represented as objects. Figure 3 shows a portion of the object class hierarchy defined in X.521 [6]. This is very similar to suggested class hierarchies outlined for network management, such as that prepared by the OSI Network Management Forum [7]. In addition to this, guidelines were given in the form of a suggested directory information tree (DIT) structure, outlining how the objects in the tree should be related. A portion of this structure is also shown in Fig. 3.

In order to incorporate authorization control into the QUIPU directory implementation [8], a new object class called QuipuObject was defined, which describes a class that contained an access control list (ACL). Any object which inherits from this will also contain an ACL, allowing access to it to be controlled. In this way, the new feature of access control was incorporated into the directory framework without necessitating any changes to the access protocol. Furthermore, the QUIPU implementation can interoperate with others that do not support access lists. The access list is made up of a SET of entries, each consisting of a trio: WHAT : WHO : ACCESS-CATEGORY.

- WHAT specifies what the ACL refers to — this can be either the entire entry, a specific attribute, or the children of a node in the DIT.
- WHO describes the entities to whom this access mode applies. It can be specified as a group of distinguished names. a distinguished name prefix (e.g., to give access to all persons within a given organization), the distinguished name to which the object itself refers, and all others.
- ACCESS-CATEGORY specifies the type of access permitted. This includes: compare, read, add, write, detect, and none.

These elements of this access control list have been selected with the X.500 directory application as a target. They can be adapted to the network management application area by taking into account the elements of service inherent in CMIS. Each element of the trio is now discussed in turn.

WHAT — Since an MIB is structured in a similar way to a DIT, the form of this part of the trio does not need to be altered.

WHO — One of the categories of user referred to in this element in the trio, is that referred to by the object itself. This is clearly applicable only to the directory, and should be omitted from a management version of this element. The ability to specify individual distinguished names, and prefixes is valuable in the management context, and should be retained, as should the other category.

In designing the security mechanism for the directory, no provision was made to designate a user as be- longing to a privileged category (e.g., SYSTEM, OPERATOR).This was not feasible in the directory context, without having the mapping from distinguished names to privileged attributes replicated in every directory service agent (DSA) participating in the directory. No such constraint exists in the management context, and the mapping of distinguished names to privileged attributes could be co-resident with the MIB.The WHO section of the ACL can thus be extended to include arbitrary user categories.

ACCESS-CATEGORY — The list of access categories specified for the X.500 directory is derived from the types of operations present in that service. A set of categories for management applications should include:

- GET — the ability to perform a get operation.
- SET — the ability to perform a set operation.
- DETECT — the presence of this access category will determine whether an attempt to access that attribute will result in a noSuchObject or an accessDenied error.
- ACTION — determines whether a specified action can be performed on the object.
- CREATE — controls the creation of new objects as children of an entry.
- DELETE— controls the deletion of a single object, or all children of an entry.
- NONE.

Summary

Authorization control can be achieved for access to MIBs by using a similar ACL implementation as that used in the QUIPU Directory implementation. With suitable modifications as outlined above, this approach provides a comprehensive means of controlling access to the MIB by authenticated users, with a minimum impact on other elements of the management framework.

Conclusion

In this article, I have outlined how the use of public key cryptosystems can be combined with a modified association control mechanism to provide peer-entity authentication and non-repudiation between two management entities. I have further explored how encryption keys can be set on a per session basis to ensure data confidentiality and, when used in association with message authentication codes, and can provide protection against replay attacks or messages being tampered with.

This article has described a modified version of X.500 access control lists that can ensure adequate authorization control over an entities access to the MIB after authentication has taken place. The use of all of these mechanisms in combination can be used to ensure a secure management service that is resistant to many forms of attack.

■ ■ ■ ■ ■

The mechanisms described here can ensure a secure management service resistant to many forms of attack.

References

[1] ECMA, Security in Open Systems : A Security Framework. Technical Report TR/46, European Computer Manufacturers Association, 114 Rue du Rhone, 1024 Geneva, Switzerland, July 1988.

[2] ISO, Information Processing Systems, Common Management Information Service Definition - Part 2, Technical Report ISO DP 9595-2, ISO, 1989.

[3] A. T. Karila, Open Systems Security — an Architectural Framework. Technical report, Telecom Finland, Business Systems R&D, P.O. Box 140,00511 Helsinki, Finland, June 1991.

[4] ISO, Information Processing Systems, Management Information Protocol Specification - Part 2: Common Management Information Protocol. Technical Report ISO DP 9596-2, ISO, 1989.

[5] K. Nakao and K. Suzuki, Proposal on a Secure Communications Service Element (SCSE) in the OSI Application Layer, *IEEE J. on Sel. Areas in Commun.*, vol. 7, no. 4, pp. 505-516, May 1989.

[6] CCITT, X.521: The Directory — Selected Object Classes, CCITT Blue Book, 1988.

[7] OSI/Network Management Forum, Forum Library of Managed Objects Classes, Name Bindings and Attributes, Technical Report FORUM 006, OSI/NMF, 1990.

[8] S. E. Kille, The Design of QUIPU (Version 2). Research Note RN/89/19, Department of Computer Science, University College London, March 1988.

Biography

DONAL O'MAHONY received B.A., B.A.I., and Ph.D. degrees from Trinity College Dublin, Ireland. After a brief career in industry at SORD Data Systems in Tokyo and IBM in Dublin, he joined Trinity College as a lecturer in computer science in 1984. He is co-author of *Local Area Networks and Their Applications* published by Prentice-Hall. At Trinity, he coordinates a research group working in the areas of Networks and Telecommunications. Within this group, projects are ongoing in X.500, Electronic Data Interchange (EDI), networked multi-media data streams, and network security. His e-mail address is: Donal.OMahony@cs.tcd.ie.