



H. Abdelkrim



L. Ballester



M. Clamagirand



G. Fouquet



B. Marquet

Security and distributed architectures

> With the advent of open, distributed IT systems, security has become crucial.

Introduction

The rapidly growing convergence of conventional telecommunication networks and the Internet, from shared management tools through to physical equipment, is leading operators to demand greater security in their networks. Thus it seems worthwhile to apply security protection to systems for which the behavior, architecture and communication protocols are already clearly defined. The framework provided by the Alcatel Management Platform (ALMAP) seems ideal

[1]. This software platform provides a means of developing Telecommunication Management Networks (TMN) [2], which are prime examples of distributed applications, and is designed for use in diverse environments.

A study of TMNs is a good way to look at security in distributed architectures. This article describes the work the Alcatel Corporate Research Center (CRC) has undertaken with the Network Application Division (NAD) on ALMAP security and how it extends to the design of the Artemis generic virtual security server.

ALMAP Security

Target of Evaluation: A Typical TMN

The security process begins with an analysis of the target, that is, of the system for which security is being assessed. This target is not ALMAP, but a TMN with the same architecture, and the same main resources and weaknesses as existing TMNs developed with ALMAP.

The TMN (see *Figure 1*) extends over the first three layers (network, element and network element) of the network management architecture, leaving aside the service and business layers. The function of the network layer is to manage the telecommunication network through the element layer, which in turn manages the network element layer containing the physical equipment.

This management architecture works according to the manager/agent model. Typically, the manager orders the agent to perform management operations. Also, the agent can use alarms to spontaneously report problems at the lower levels to the manager. The manager, another of ALMAP's interfaces, is an application front-end (i.e. man/machine interface), which can access the other application front-ends via a "navigation" mechanism. The aim is to link each management operation with the appropriate front-end.

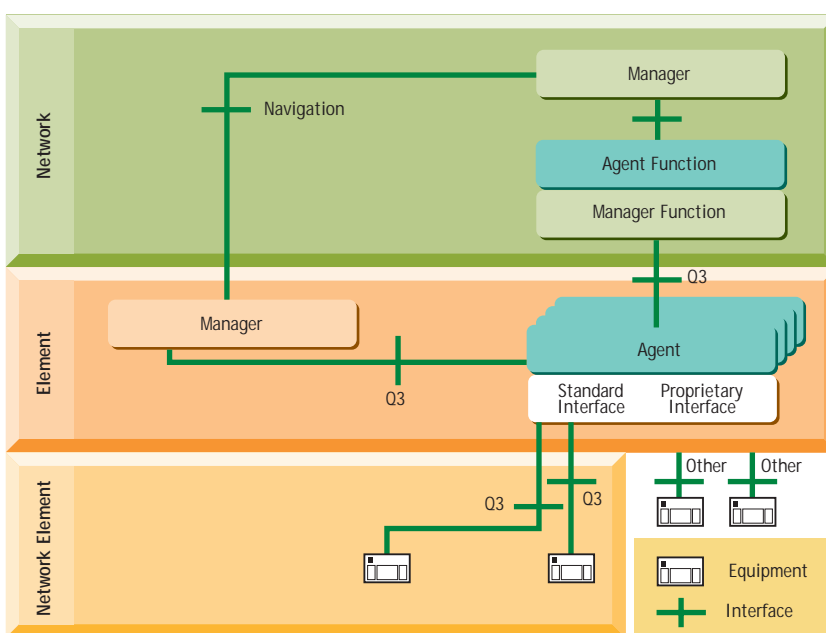


Figure 1 – TMN layers

From the security viewpoint, this system (see *Figure 1*) is typical of a TMN developed with ALMAP, for the following main reasons:

- A management operation can involve several components.
- The main communication protocols between the components (management and navigation) are in place.
- Access to the manager is all that is needed to access the resources (e.g. telecommunication equipment).

Analysis of the Target of Evaluation

The aim of a risk analysis is to rigorously determine the objectives of the security process. To achieve this, the target of evaluation is studied to determine its weaknesses and the threats it faces. Based on this analysis and the customer's requirements, the objectives can be determined.

Vulnerability

A TMN is a distributed architecture and, as such, is prone to the same weaknesses as telecommunication network management:

- Access to the manager allows access to the managed resources.
- Agent application cannot identify the origin of the management command.
- Communication links between agents and manager are not protected.

Threats

The main threats to the target of evaluation are that:

- Important information about the telecommunication network might be disclosed or used to impair network operation.
- Unauthorized user might gain access to the resources, manipulate them and thereby present a threat to network operation.
- User might usurp the identity of another and access the network with the same privileges as an authorized operator.

- Modifications might be introduced into the information being transferred, or into the databases, and impair network operation.
- Unauthorized person might modify stored information, databases or information being transferred in communications between the various components.
- During authorized access, resources might be diverted from their intended use; for example, a key intended to sign documents could be used to cipher communications.
- A resource might refuse a service to a duly authorized user because, for example, an intruder has overloaded it with spurious enquiries, preventing it from performing its function.

Security objectives

The aims of the security process are based on a risk analysis of the security objectives and what the customer needs in terms of security. The management platform needs to be able to:

- Guarantee the confidentiality and integrity of information saved or communicated.
- Guarantee the availability of the information and applications.
- Prevent an operator from denying responsibility for its actions and their effects.

The choice of objectives is intended to cover a large number of requirements, given that, depending on the TMN to be protected and its environment, they will be achieved to varying degrees. For example, the integrity of information could be disregarded if there is no possibility of anyone corrupting the data, or if corruption would not affect TMN operation.

A Solution

One security solution that satisfies the above objectives is to define a security policy and a set of

resources, which we call "services", with which to apply the policy.

Given that security concerns all the components of a system, it is important to define the scope within which ALMAP will be able to operate to protect a TMN before defining the countermeasures. ALMAP can be used to produce TMNs for several environments. A finite number of target environments were selected, and were described in terms of the hardware and basic software (operating system, network protocols, tools). While this approach makes it possible to cover the wide variety of environments for the development of ALMAP software, it is limited with respect to security because it only reflects part of the final TMN. In practice, some of the weaknesses exploited by pirates relate to specific component versions of certain operating systems. Also, not all the checks are accessible to the software developed within the paradigm proposed by ALMAP.

For example, a system operator whose Unix system has been breached will have a duty to rectify the Unix vulnerabilities exploited by the intruder. The operator can review the known weaknesses and strengthen them using available patches or configure them securely. In addition, the operator can make use of strongly protected Unix versions, like Trusted Solaris, for critical elements in the TMN. In both cases, the TMN needs the operator to undertake any such action.

Of course, it is up to ALMAP to indicate the critical points, but ALMAP cannot force the operator to protect these points. Physically, all it can do is make recommendations for achieving the required level of security. Anecdote has it that Windows NT 3.1 was certified E3-FC2 according to Information Technologies Security Evaluation Criteria (ITSEC), provided that the floppy drive was *physically* deactivated [3].

Consequently, the operator must be provided with a list of countermeasures and the target environ-

ment must be accorded a degree of trust. TMN security is the responsibility of the environment administrator, since it involves installing patches as they become available, and configuring a Virtual Private Network (VPN) that is both solid and impenetrable. The scope of the security aspect considered here stops at the infrastructure on which the TMN is deployed. The operating system is assumed to offer a minimum of security (as do ALMAP target systems: HP/UX, Sun/Solaris, Microsoft Windows NT) and the network is assumed to be isolated and protected by, for example, Transmission Control Protocol wrappers *plus* Secured Shell (SSH) or IPsec. The issue is then security within the system.

Security Policy

The security policy defines the way in which the security services are used to achieve the chosen objectives. The general policy that we propose to achieve the security objectives is as follows:

- Each operator or application (e.g. agent) involved in a secured management process must be authenticated.
- Access control must govern entry to each application. To ensure this, each operator is assigned a role defining which applications it can access (e.g. operator, administrator).
- Access to the resources must be controlled.
- Communication channels that

support management exchanges on the TMN must be protected by using the secured exchange service.

This security policy can be used typically in security management to ensure that only an administrator can access the central security administration function.

The policy is general enough to cover a large number of requirements. For a given TMN, it can be refined by detailing how the security services will be used and extending them where necessary.

Security Services

ALMAP's existing SEC component [1] provides a part of the access control service mentioned above. It controls access to the Human/Machine Interface (HMI) menus and the managed objects. The term "managed object" is used, for example, to mean the representation of a TMN network element.

A new SEC^E component handles other services. More specifically, SEC^E is a set of security services for:

- Authenticating a user or an application – typically, an agent.
- Controlling access to applications. The first level of access control is encountered on entering the application; the second is handled by the SEC component.
- Securing exchanges, typically between a manager and an agent, that is, authenticating the

communicating entities, ciphering the data and checking its integrity.

- Guaranteeing the confidentiality and integrity of data stored in memory or a file.
- Storing information by which users can be held accountable for their actions.

When used together, SEC^E and SEC constitute a solution that completely secures the target of evaluation (see *Figure 2*).

Implementing the Solution

The security services have been implemented according to the above policy to protect a single management session involving just one manager controlling just one agent.

The simplified representation of the architecture in *Figure 3* shows how SEC^E is attached to applications, and the steps involved in opening a secured session:

1. The agent application is activated, following which it is authenticated by the authentication service. (Note: In an operational context, it is commonplace for an agent to delegate a part of its management to another agent. Authentication of these applications is essential to ensure that a pirate agent cannot control another agent in the TMN.)
2. User supplies the login front-end with his or her name and password and then chooses a role. The login front-end authenticates this information via the authentication service.
3. After authentication, the front-end allows the user access to the manager, which then checks whether the user is allowed access by scanning the list of privileges broadcast by the access control service.
4. Then, if everything checks out from a security point of view, the manager application opens a management session with the agent. Using the secured exchange service (which was

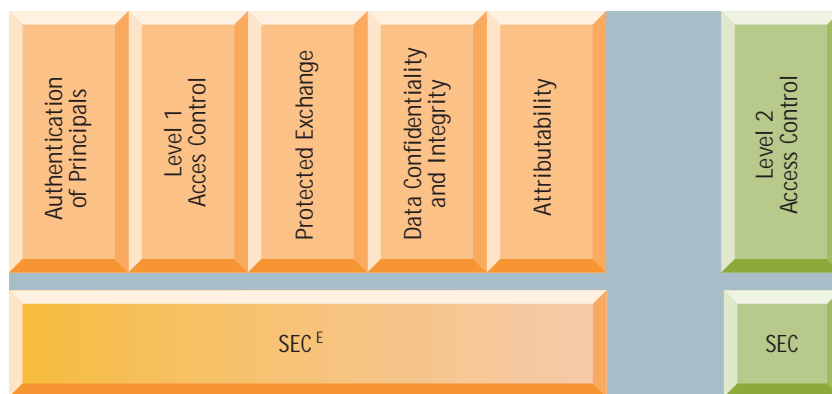


Figure 2 – Security services

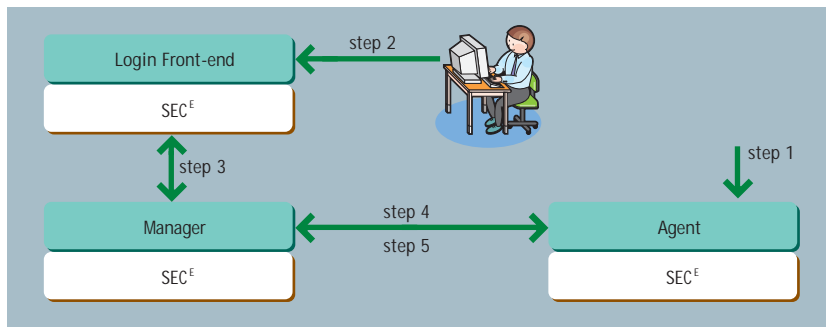


Figure 3 – Implementation of the solution

responsible for transparently transferring the security information concerning the user from manager to agent), the agent uses the access control service to check that the user exists and is allowed access.

- Finally, the management session can begin with the necessary confidentiality and integrity provided by the secured exchange service.

Cost of integration

The above scenario illustrates the close interrelationship between the management applications and the security of SEC^E . Fortunately, SEC^E can be simply and cost-effectively integrated with the management applications to achieve this level of security. Integration involves:

- Securing the communication layer (C++ OSI Management Extended Toolchain; COMET), which is common to both manager and agent in the ALMAP architecture.
- Developing a login front-end that can invoke all the managers.
- Adding a few lines of code to the manager (see *Figure 4*) and to the agent.

Cost of use

The secured exchange service provides a useful measure when it comes to evaluating the cost of using security because it is frequently invoked. The test, the results of which are given below, was run on the SEC^E secured exchange service. For this test, the 64-bit Data Encryption System

(DES) algorithm was selected for encryption and the MD5 algorithm for the signatures.

The test shows that secured exchanges for each communication take two or three times as long as unsecured exchanges (see *Figure 5*). This slowing down of encrypted communications is inherent to the choice of encryption. Whatever solution is chosen, security has a cost, so a judicious balance between risks and countermeasures must be found.

Flexibility of the Solution

SEC^E has the advantage of being totally independent of the security technologies. In practice, each SEC^E service is designed to be generic enough to cover the wide range of security technologies that will perform this service. For example, the secured exchange service complies with the Internet standard defining the Generic Security Services API [4].

SEC^E was first implemented using SESAME [5] technology. SESAME

is a security server enhanced with numerous libraries of functions for ensuring the security of distributed applications.

In order to prove that SEC^E makes the applications independent of the security technologies, the secured exchange service was implemented on top of Secure Socket Layer / Transport Layer Security (SSL/TLS) [6], which is based on certificates (i.e. electronic documents certifying the identity of a user or application). It therefore differs fundamentally from SESAME which uses a reliable third party, a recognized third party providing guarantees as to identities.

General Roll-out

The chosen solution fits in with the “conventional” TMN framework, by making use of Q3 interfaces [7]. Currently there are plans to replace these interfaces with Common Object Request Broker Architecture (CORBA) interfaces [8] and to review the ALMAP architecture to organize it around a software bus that complies with this architecture. Given this background of migration to CORBA, it is important to note that:

- Services defined for SEC^E can be found in the specifications of CORBAssec, the CORBA security service [9].
- Their interfaces are similar to those of CORBAssec.

```
SecurityPrincipal P; // Declares the structure
                    // containing security
                    // information concerning
                    // the user

P.SetCredential(); // Recovers this information
                  // from the login
                  // front-end

P.Authorization("manager"); // Checks whether user
                              // P has access to the
                              // "manager" application

comPkg.declareOwner(& P); // Declares that P is
                          // owner of the
                          // communication channels
                          // opened by the manager
```

Figure 4 – C++ code added to the manager for its protection

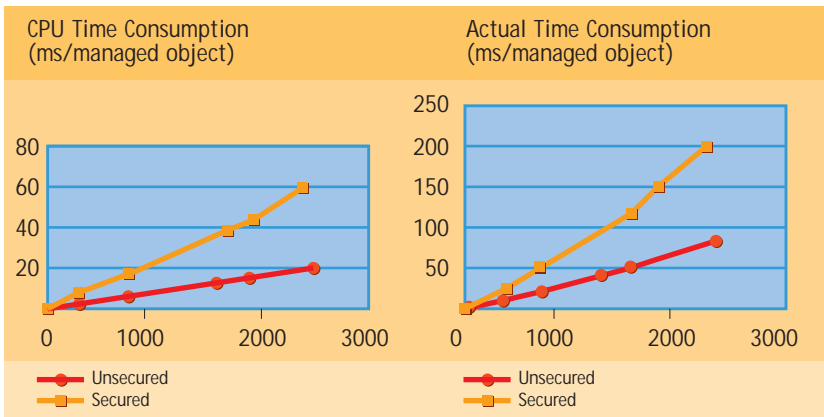


Figure 5 – Time taken for secured and unsecured management exchanges (DES + MD5, 64 bits)

- CORBAsec also raises the problems of the ease with which the safety mechanisms can be replaced and their ability to inter-work.

In short, the conclusions in terms of security for a system distributed through CORBA are similar to those concerning a TMN distributed through “conventional” means. In particular, they do not depend on the technology used for distribution.

The similarities noted above force us to consider a more general problem which, if it were to be resolved, could enable us to propose a solution for many Alcatel products. This more general problem is that of security for a telecommunication application. The planned solution is to use a security server, known as Artemis (Alcatel Response for TELEcommunication Modular and Integrated Security), which should provide the application with all the necessary security services.

Using this server, our experience with SEC^E has given us some pointers, such as the validity of a security guarantee approach. However, it does not offer an answer to issues arising from the desire to widen the project: how to model security for all target applications. What procedure can hope to provide the required degree of flexibility for the security mechanisms?

Global Approach

The Artemis project is part of a global approach that is intended to span a wide range of requirements and to be consistent with a variety of technologies and/or platforms in the application-oriented areas of telecommunications.

Generic Services

This approach is embodied in the provision of a set of generic security services. However, in many cases, such as with the CORBA security service interfaces, the generic nature means that a more complex programming interface has to be designed, thus adding to the customer's work. The ultimate aim is therefore to link a simple interface with the generic service. Simplicity can be vitally important when it comes to security. Indeed, the simpler the interface, the less the risk of compromise or misuse,

and the greater the effectiveness of the security function.

Extensive Range of Services

A simple interface must not be taken to mean a reduced interface. Because the potential customer base is so extensive and varied, an extensive and feature-rich range of services must be provided. This means using a strict methodology. In practice, it is important to ensure that an extensive range is consistent. In the case of Artemis, a security assurance approach must be followed if these guarantees are to be obtained.

Assurance Approach

Conceptually, security assurance is like quality assurance; it involves applying a proven approach and obtaining maximum assurance of the expected results. This is reflected in the effectiveness of the adopted solution and in how far it satisfies the expressed needs. Artemis is based on a new International Standards Organization (ISO) standard [10], known by the name of “Common Criteria”, which establishes a framework for the definition, development, evaluation and certification of secured solutions.

Initially, this standard is being used to express the security needs of all potential customers. Thus, one can select a set of functional security components from one of the catalogs supplied in the standard.

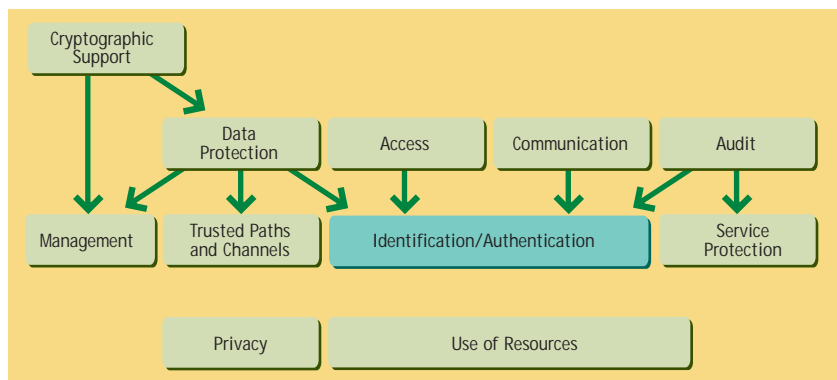


Figure 6 – Set of requirement classes with their dependencies

Consistency of Deployed Services

By selecting components from standard catalogs, consistency of requirements can be achieved. Because the choice of a component is also expressed in terms of dependencies with other components, respecting these dependencies guarantees the required consistency.

Figure 6 illustrates the dependency links between the requirements. When the secured application is deployed, these links will emerge as dependencies between software modules. The choice of components therefore also translates into a deployment architecture for security functions.

This architecture makes it possible to deploy a coherent set of services. For example, if an audit service is to be provided, it is first necessary to deploy an identification and authentication service.

This set of requirements is stored in a "security profile", which is the response to the threats to and general vulnerabilities of telecommunication applications at which the project's security services are aimed. This profile, together with an object model, provides the basis for a common security language for telecommunication applications.

Virtual Security Server

In the Artemis approach, the issue of genericity is solved by introducing virtual services. The software components are not allowed direct access to the security services, but to a software layer which calls the security function or the set of security functions needed to perform the required service.

This concept of a virtual security server enables security service calls to be unified (see Figure 7) according to requirements and not according to the technologies to be supported. For example, the applications will always call the identification and authentication service in the same way, whether

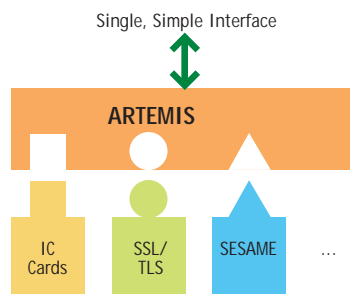


Figure 7 – Just one interface

based on CORBA type Object Request Brokers (ORB) or other Java platforms.

This concept means that there is total freedom to choose the underlying security mechanisms. It adds the flexibility required to adapt to country-specific legal requirements and to achieve the best trade-off between security guarantees and processing capabilities.

Conclusion

At present, telecommunication operators are demanding a secure infrastructure. These operators are beyond the scope of ALMAP, and they can be satisfied by compiling a list of recommendations for each target environment. However, operators will soon be looking for the type of security provided by ALMAP, and we must be ready and able to offer a response.

From the TMN developer's point of view, the solution described in this article can be incorporated at little cost. From the TMN user's viewpoint, it has an impact on encrypted communications that is intrinsic to the choice of encryption. This reminds us that, whatever solution is adopted, security has a cost, so a judicious balance between risks and countermeasures must be found. Ultimately, the main job is to design and implement a solution like SEC^E in ALMAP.

Besides securing ALMAP, SEC^E demonstrates the viability of a flexible solution which accepts

miscellaneous security mechanisms and therefore can be adapted to different laws and technologies. Also, the similarity between this solution and solutions defined in a much wider framework gives us reason to believe that, if we deal with the general problem of security in distributed architectures, we can achieve a solution that could apply to many Alcatel products. This is the objective of the Artemis project. ■

References

- 1 ALMAP: <http://www.alcatel.com/telecom/mbd/products/products/detailed/almap/framewor.htm>.
- 2 ITU-T, M3010: Principles for a Telecommunications Management Network, 05/1996.
- 3 ITSEC: Information Technologies Security Evaluation Criteria, v1.2, 06/1991.
- 4 IETF: "Generic Security Services API", RFC 2078.
- 5 SESAME: <http://www.esat.kuleuven.ac.de/cosic/sesame>.
- 6 IETF: "Transport Layer Security", RFC 2246.
- 7 ITU-T: "Lower and Upper Layer Protocol Profiles for the Q3 Interface", Recommendations Q.811 and Q.812, 1993.
- 8 OMG: "Common Object Request Broker Architecture Specification", Rev. 2.3, 01/12/1998.
- 9 OMG: "CORBA Security", Chapter 15 of "CORBA services specification", Rev 1.2, 17/12/1999.
- 10 ISO/IEC 15408: "Evaluation Criteria for Information Technology Security", 1999.

Hanine Abdelkrim is a Research Engineer responsible for ALMAP product risk analysis and development of the SEC^E component at the Alcatel Corporate Research Center in Marcoussis, France.

Laurent Ballester is a Research Engineer responsible for the Artemis specification at the Alcatel Corporate Research Center in Marcoussis, France.

Michel Clamagirand is in charge of the security study, responsible for development specifications and follow-up at the Alcatel Corporate Research Center in Marcoussis, France.

Guy Fouquet is Group Leader, responsible for maintaining contact with the Operational Divisions on security matters, at the Alcatel Corporate Research Center in Marcoussis, France.

Bertrand Marquet is a Research Engineer for Artemis specification at the Alcatel Corporate Research Center in Marcoussis, France.