

Securing a real-time field area network using smart cards

by

Gerhard Petrus Hancke

Submitted as partial fulfillment of the requirements for the degree
Master of Engineering (Computer Engineering)

in the

Faculty of Engineering

UNIVERSITY OF PRETORIA

August 2003

Keywords: smart cards, real-time, field area network, security, authentication, authorization, access control, smart cards, real-time, field area network, security, authentication, authorization, access control.

Summary

Field area networks are rapidly expanding to include a wide range of applications. Intelligent nodes on the network will be installed in a small to medium geographical area to monitor and control processes. Such nodes are generally connected to a centralized gateway used by a service provider to monitor and control various applications. The growth in popularity of ubiquitous computing requires the use of embedded network processors in everyday objects. Even though the idea of interaction between the digital devices around us could bring a great deal of convenience it also introduces great risks. Therefore such applications would not only require measurement, control and communication functionality but also a high level of security.

Smart cards offer a simple, inexpensive method of incorporating a cryptographic processor into an embedded system that will allow for the implementation of security services. A field area network has resource limitations that influence security service implementation, such as low bandwidth, limited processing power, limited storage capacity and limited communication protocols.

This dissertation discusses the implementation of a security policy for embedded field area networks used in distributed real-time applications, using smart card technology. The primary objective is to formulate a policy that can be implemented to secure a field area network. The secondary objective is to determine whether this policy can be implemented using mechanisms provided by smart card technology, while maintaining reasonable system performance. It states the approach taken to finding a viable solution to the problem defined above. A comprehensive literature study provides background on relevant technology and possible solutions. In a system overview the system's boundaries and functional requirements are defined. The implementation section outlines possible solutions and describes how these can be implemented. Evaluation, verification and quantification of the performance of the proposed system are performed according to the experimental procedures described. The results obtained are documented and discussed. In the conclusion the proposed solution and the findings from the results are placed in context. Future topics of research in this field are suggested.

Keywords: cryptography, field area networks, smart cards, information security

List of Abbreviations

- AES - Advanced Encryption Standard
API - Application Programmers Interface
BMA - British Medical Association
CA - Certification Authority
CLEF - Commercial Licensed Evaluation Facilities
COBIT - Control Objectives for Information and Related Technologies
COS - Card Operating Systems
CTPEC - Canadian Products Evaluation Criteria
CRAMM - UK Government's Risk Analysis and Management Method
DES - Data Encryption Standard
DF - Dedicated Files
DSA - Data Signature Algorithm
DSS - Data Signature Standard
EEPROM - Electrically Erasable Programmable Memory
EF - Elementary Files
FAN - Field Area Network
FIPS - Federal Information Processing Standards Publications
GMITS - Guidelines for the management of IT security
IETF - Internet Engineering Task Force
ITSEC - Information Technology Security Evaluation Criteria
ITU-T - International Telecommunication Union Standardization Sector
IVV - Independent Verification and Validation
MAC - Message Authentication Code
MASC - Multi-Application Smart Card
MD - Message Digest
MOAS - Multi-Application Operating Systems
OP - Open Platform
OSI - Open System Interconnection
PDA - Personal Digital Assistant
PIN - Personal Identification Number
PKI - Public Key Infrastructure
QoS - Quality of Service
RAM - Random Access Memory

- RFC - Request for Comments
- ROM - Read Only Memory
- RSA - Rivest Shamir Adelman
- SHA - Secure Hash Standard
- SP - Service Provider
- TCP/IP - Transfer Control Protocol/Internet Protocol
- TOE - Target of Evaluation
- VM - Virtual Machine
- VOP - VISA Open Platform

Contents

1	Research Overview	1
1.1	Introduction	2
1.2	Problem Statement	4
1.3	Scope	4
1.4	Research Context	5
1.5	Research Objectives	5
1.6	Research Approach	7
1.6.1	Research Method	8
1.7	Document Overview	9
2	Literature Study	11
2.1	Overview and Related Work	12
2.2	Information Security	13
2.2.1	Security Mechanisms	15
2.2.2	Security Threats	16
2.2.3	Security Services	18
2.2.4	Security Assurance	30
2.2.5	Key Management	32
2.3	Distributed Field Area Networks	33
2.3.1	Security in Embedded Systems	36
2.4	Smart Card Technology	37
2.4.1	Smart Card Architecture	37
2.4.2	Smart Card Operating Systems	45
2.4.3	Why Use Smart Cards?	54
3	System Overview	56
3.1	System Definition	57

3.1.1	System Architecture	58
3.1.2	Assumptions	58
3.2	Risk Analysis	60
3.2.1	Identification of Assets	60
3.2.2	System Requirements	61
3.2.3	Vulnerabilities and Threats	62
4	System Specifications	65
4.1	Security Policy Overview	66
4.2	System Components	66
4.2.1	Node	67
4.2.2	Private Network	69
4.2.3	Public Network	70
4.2.4	Gateway	71
4.2.5	Owner	72
5	Implementation	74
5.1	Security Services	75
5.1.1	Availability	75
5.1.2	Authentication	75
5.1.3	Access Control	76
5.1.4	Integrity and Non-repudiation	77
5.1.5	Confidentiality	78
5.1.6	Key Management	78
5.2	System Components	80
5.2.1	Node	80
5.2.2	Private Network	82
5.2.3	Public Network	83
5.2.4	Gateway	83
5.2.5	Owner	85
5.3	Security Policy Overview	86
5.3.1	Node Registration	87
5.3.2	Owner \Rightarrow Node	90
5.3.3	Node \Rightarrow Owner	93
5.3.4	Node \Leftrightarrow Node: Trusted Gateway	97

5.3.5	Node \Leftrightarrow Node: Untrusted Gateway	100
5.3.6	Node \Leftrightarrow Node: Direct	103
5.3.7	Key Management	104
5.3.8	Additional Considerations	104
5.4	Mechanism implementation	104
5.4.1	Confidentiality	107
5.4.2	Digital Signature: PKI	107
5.4.3	MAC: Symmetric encryption	108
5.4.4	Authentication: PKI	108
5.4.5	Authentication: Symmetric	109
5.4.6	Messages	109
6	System Evaluation	110
6.1	Security Assurance	111
6.2	Performance of Security Mechanisms	112
6.3	Test System	113
7	Results	116
7.1	Security Assurance	117
7.2	Performance of Security Mechanisms	120
7.2.1	Confidentiality	121
7.2.2	Digital Signatures	121
7.2.3	Authentication: PKI	121
7.2.4	Authentication: Symmetric	125
7.2.5	Messages	126
8	Conclusion	127
8.1	Summary of the Work	128
8.2	Summary of the Results	128
8.2.1	Effectiveness of Security Policies	128
8.2.2	Comparison of Security Policies	129
8.3	Conclusions	129
8.4	Suggestions for Future Work	130
9	References	131

RESEARCH

Chapter 1

RESEARCH OVERVIEW

1.1 INTRODUCTION

The growth in popularity of ubiquitous computing requires the use of embedded network processors in everyday objects. Even though the idea of interaction between the digital devices around us could bring a great deal of convenience it also introduces great risks.

Field area networks are rapidly expanding to include a wide range of applications [1]. Embedded intelligent nodes on the network are generally connected to a centralized gateway used by a service provider to monitor and control various applications. Field area networks have previously been implemented as autonomous systems as shown in figure 1.1, therefore security has never been a great concern.

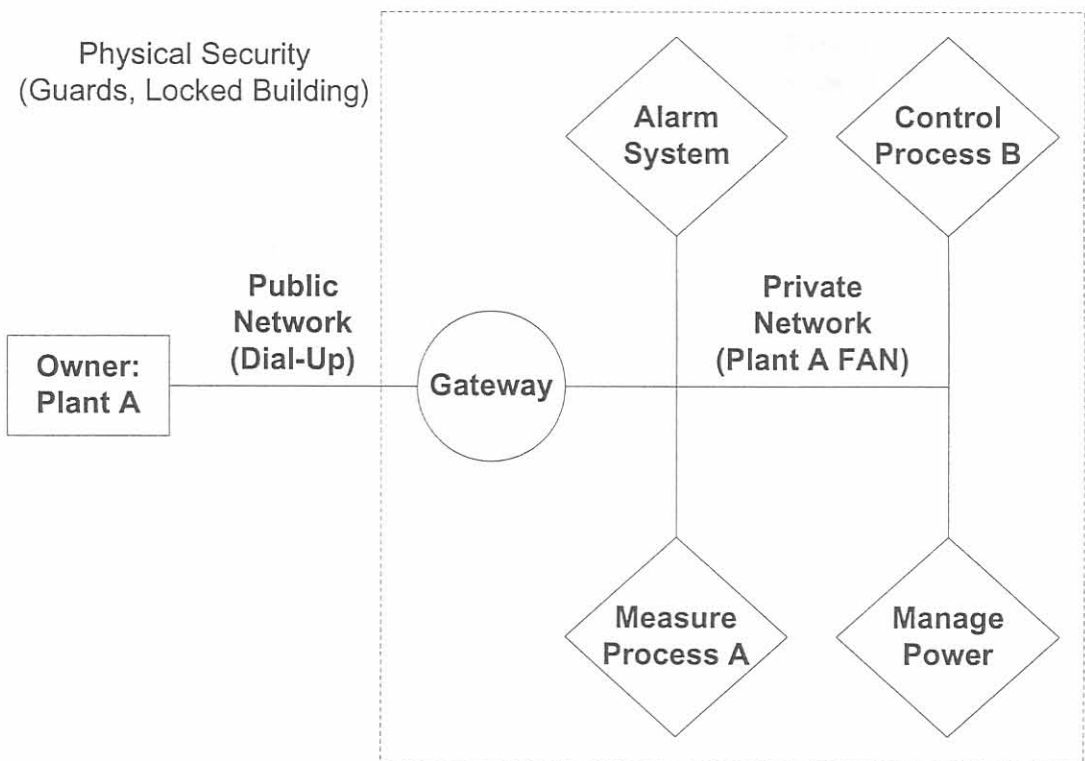


Figure 1.1:
Autonomous field area network

These days entities from different vendors, or with different owners, may be connected to the same network and therefore networks are no longer guaranteed to be autonomous. Embedded intelligent nodes on the network can be installed in a small to medium geographical area (e.g. an office, a group of flats or a factory) as is shown in figure 1.2. Each entity needs a secure way to relate to its owner, vendor and peers in the network without being compromised [2].

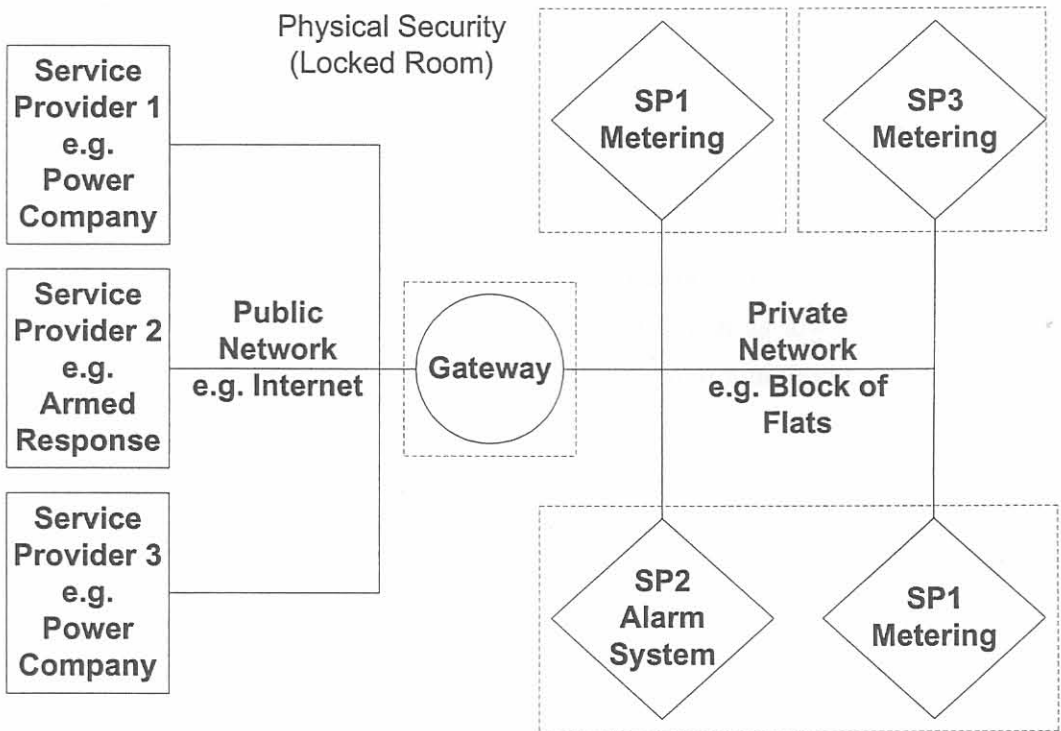


Figure 1.2:
Non-autonomous field area network

Some new applications such as power metering or prepaid systems require that the data on the network be secured. In these cases the users, or unauthorized third parties, cannot be allowed to have the means to possibly manipulate such data. For such applications the client would not only require measurement, control and communication functionality but also a high level of security. The node must therefore be able to capture, manipulate and secure data and send it to the gateway, from where it is presented to the service provider. To physically secure the entire system like the previous autonomous model would be im-

possible. The gateway and the nodes can be housed in secure containers or locked rooms, but the networking infrastructure would be vulnerable to intrusion. The gateway is also more vulnerable to attacks from the public network as it can now be accessed by multiple users.

Smart cards offer a simple, inexpensive method of incorporating a cryptographic processor into an embedded system that will allow for the implementation of security services. This dissertation discusses the implementation of a security policy for embedded networks, using smart card technology. This chapter defines the problem statement, research objectives, research approach, context and scope of the work undertaken.

1.2 PROBLEM STATEMENT

The real problem to be addressed is the implementation of security services, using smart cards, in a field area network with resource limitations such as low bandwidth, limited processing power, limited storage capacity and limited communication protocols.

The challenge would be in identifying the inherent risk and threats for embedded networks in context of possible applications for these systems. A novel security policy must then be formulated to address information security within field area networks. The proposed security policy must ensure that the confidentiality, integrity, availability and authentication services are upheld. All security services must be implemented using only mechanisms provided by smart card technology.

1.3 SCOPE

The scope of the research is to propose a security policy, using smart card technology, that will provide security services to an embedded field area network used in distributed real-time applications. Performance issues related to the implementation of security mechanisms using smart cards are also addressed. It does not focus on distributed real-time programming, network architecture or communication protocol issues beyond those required to implement a sound security policy. The security policy incorporates accepted security mechanisms and does not introduce new cryptography techniques.

1.4 RESEARCH CONTEXT

A field area network offers a reliable solution for critical real-time systems functioning in a harsh environment. Separate systems have been developed to incorporate the functionality of FAN's [3],[4] and to take advantage of the security features offered by smart card technology [5],[6],[7]. The two technologies have never been combined to provide secure networking services. FANs have previously been implemented as autonomous systems, and therefore security has never been a great concern. These systems can be applied in home automation, prepaid systems, real-time distributed systems and e-commerce (the user could request a service online and an enabling token passed through the system connected to his/her TV, Hi-Fi, etc.). For these networks to be a commercially viable proposition a range of security services must be provided in order to protect the system from malicious activity [8]. Smartcards might provide a viable way of providing the security functions needed to implement a sound security policy, which will allow field area network applications to expand significantly. A few applications that will be made possible are:

- prepaid network systems for commercial services,
- secure data logging applications (power consumption metering),
- remote control & measurement of critical processes,
- secure home or office automation and
- security application such as access control to a building.

1.5 RESEARCH OBJECTIVES

The main objective would be to formulate a policy which could be implemented successfully into a field area network, using the security mechanisms provided by a smart card, while maintaining reasonable system performance. The secondary objective will be to determine how the security mechanisms provided by smart card technology can be applied to secure an embedded network. To achieve this, a number of sub-objectives need to be accomplished. These sub-objectives are:

- Field area network implementations must be analysed and the way in which various entities interact studied. The main assets of the network system must be identified in order to determine which entities are critical to the overall operation of the system.

Once the important entities have been identified and the operation of the system studied a list of functional requirements must be drawn up. These are the operations that must be performed by the network entities in order for the system to function correctly. Vulnerabilities and possible threats to the system assets and functional requirements should then be determined.

- Security policy implementations in other types of networks must be analysed. Possible security services and mechanisms used in implementing these security policies must be investigated. It should also be noted which services address which threats and vulnerabilities.
- Develop a security policy which will mitigate threats and enforce the functional requirements. Determine the security services which are required to enforce the security policy. It must be determined which mechanisms must be combined to implement the security services required by the policy. In order for this policy to be successful a set of specifications for each network component needs to be formulated. These specifications give guidelines of which mechanism needs to be implemented where and what information needs to be communicated and stored.
- Smart card technology must be studied and the different security mechanisms provided investigated. Different means to implement the required services using these mechanisms must be proposed. These mechanisms must be successfully implemented in order to test performance and ensure functionality.
- Criteria must be defined in order to evaluate the overall security policy. Assurance must be provided that the policy is effective and that system vulnerabilities and threats are addressed. A benchmark needs to be defined to measure the performance and security level of individual mechanisms used to provide security services. This should then be used to compare different service implementations in order to determine which are best suitable to secure an embedded network.

1.6 RESEARCH APPROACH

A number of questions were identified to assist in better defining and solving the problem at hand:

The operation of current networks and how functionality is provided need to be researched, including hardware and resource considerations, the requirements for such a network to ensure functionality, how these functional requirements can be disrupted and how these functional requirements and assets can be protected. This will include an investigation into the critical components of such networks, whether these components are vulnerable and how these components are threatened, the services generally provided to mitigate these threats and what role each component plays in the success of the policy.

Security mechanisms to provide the necessary security services needed to be defined, including issues such as the mechanisms provided by smart cards, how these mechanisms can be used to implement the security services required, whether these mechanisms can successfully be implemented into a working system, how the performance of different mechanisms can be measured and compared, the performance of the implemented mechanisms, and whether the security mechanisms provided by the smart cards can secure the network while ensuring reasonable network performance.

How can it be assured that the formulated security policy is feasible for embedded field area networks, as well as secure and effective in addressing all vulnerabilities?

1.6.1 Research Method

1. Literature Study

- The scope of the work is further defined.
- Relevant work from other sources are identified.
- General information security principles are discussed.
- Available security services and mechanisms are documented.
- Common threats and vulnerabilities are documented.
- The design considerations and architecture of distributed field area network are discussed.
- Different smart card technologies are identified.
- Mechanisms provided by smart cards are studied.
- The advantages and disadvantages of smart card technologies are discussed.
- Industry standards addressing security assurance for security policies are discussed.

2. Problem Analysis

- Field area network implementations are analysed.
- A network system is defined to model a field area network.
- Assets and functional requirements are determined.
- Possible vulnerabilities and threats to the system are identified.
- Risk analysis is performed to determine the significance of the vulnerabilities and threats.
- Guidelines are given for the functions required by each component in the network.
- Special hardware requirements of embedded networks are determined.
- The security services needed to secure the system are identified.

3. Design

- A security policy that addresses the vulnerabilities and threats is defined.
- Mechanism needed to provide the necessary services are determined.

- A communication protocol implementing the 3rd and 4th layer of the OSI layer is suggested.
- Methods of implementing the required mechanisms using smart cards are proposed.
- Determine evaluation criteria for the security policy
- Determine benchmarks for evaluating system performance and performance of implemented mechanisms.

4. Implementation

- Detail is given of how the services will be implemented using smart cards.
- The security mechanisms are implemented on smart cards.
- A test system is build and documented.
- Security services are implemented in the system using smart card mechanisms.

5. Review and Assessment

- Using the test system collect data on the performance of the individual services and mechanisms.
- Evaluate these mechanisms and system performance by comparing results to the benchmarking criteria.
- Assurance is provided that the proposed security policy is secure and addresses all identified threats and vulnerabilities.

1.7 DOCUMENT OVERVIEW

This dissertation consists of the following sections:

Chapter 1 (Research Overview):

Describes the problem and the research approach to finding a viable solution.

Chapter 2 (Literature Study):

Provides background information on relevant technology and possible solutions.

Chapter 3 (System Overview):

Defines the system's boundaries and functional requirements.

Chapter 4 (System Specifications):

Identifies aspects that the solution must address.

Chapter 5 (Implementation):

Outlines possible solutions and describes how they can be implemented.

Chapter 6 (System Testing):

Shows the experimental procedures used to verify and quantify the performance of the proposed implementations.

Chapter 7 (Results):

Documents and explains the results achieved during testing.

Chapter 8 (Conclusion):

Places the proposed solution and the findings from the results in context. Also suggests some further topics of research in this field.

Chapter 2

LITERATURE STUDY

2.1 OVERVIEW AND RELATED WORK

As the costs of high-end computer equipment get less the networking community is moving away from the traditional server-client and autonomous networking models. The sharing of resources, as a result of distributed computing, has changed the nature of network applications. Information is no longer housed at a central secured location, therefore network security has become the most important aspect in information networks today. Bandwidth requirements, the range of services and QoS are all necessary requirements for a functional network but a security failure is the most catastrophic and almost tend to negate the other requirements.

Field area networks are rapidly expanding to include a wide range of applications [1]. Embedded intelligent nodes on the network can be installed in a small to medium geographical area (e.g. an office, a group of flats or a factory) to monitor and control real-time processes. Such nodes are generally connected to a centralized gateway used by a service provider to monitor and control various applications. For such applications the client would not only require measurement, control and communication functionality but also a high level of security. The node must therefore be able to capture, manipulate and secure data and send it to the gateway, from where it is presented to the service provider. Field area and automation networks have mainly been used as stand-alone systems. With these technologies gaining in popularity and being used in new applications they become more distributed. This introduces new challenges and considerations in terms of security. In the power measurement/billing application the person with access to the node is not the owner but in fact the client. In addition the network runs through an entire apartment complex with a few nodes per residence. In this case the residents would be able to change either their own or their neighbours' readings.

The growth in popularity of ubiquitous computing requires the use of embedded network processors in everyday objects. Even though the idea of interaction between the digital devices around us could bring a great deal of convenience it also introduces great risks [9]. In the past much emphasis has been put on information security for high performance computers and data networks. These methods suffice for information systems incorporating powerful processing capabilities. The embedded and real-time nature of the network itself poses problems other than encountered in the networked PC environment. Nodes could be weak in storage or processing capability due to size or power concerns with real-

time operation placing constraints on the time that can be allocated to security overhead. As a result security technology used in the PC/Internet environment or developed with TCP/IP in mind will not be practical. The traditional security model identifies three main classes [10]:

- Confidentiality: Information not divulged to wrongful entity.
- Integrity: Information cannot be maliciously modified.
- Availability: The system cannot be prevented from performing its functions.

These classes are however all dependant on identification, authentication and authorization principles to be successful. A failure to authenticate correctly will easily lead to breaches in the rest of the security policy. Smart cards offer security mechanisms to provides security services under various circumstances [11], [12].

Entities from different vendors, or with different owners may be connected to the same network and therefore networks are no longer guaranteed to be autonomous and secure. Each entity needs a secure way to relate to its owner, vendor and peers in the network without being compromised. For distributed networks to be a commercially viable proposition a range of security services must be provided in order to protect the system from wrongful activity. Smart cards offer a simple, inexpensive method of incorporating a cryptographic processor into an embedded system. Each smartcard has a cryptographic processor that performs the algorithms used to lock, unlock and verify data. The smart card is also strongly protected against different security attacks [13]. These two technologies together could very much benefit everyday applications [5].

After studying the available literature it would appear that little or no work has been done on securing field area networks. This research dissertation will be one of the first to document information security principles and performance for embedded field area networks. An abundance of information is available on security methodology, smart card technology and distributed field area networks.

2.2 INFORMATION SECURITY

Information security is a field of study that is enjoying tremendous attention in computing communities. It can be described as an ever-growing technology and many organizations

are investing heavily to develop security features. There are three definable aspects to a secure system [14]:

- Threats: A potential for violation of security, which exists when there is a circumstance capability, action, or event that could breach security and cause harm. A security attack is a threat that has realised.
- Security mechanisms: An action that is designed to detect, prevent, or recover from a security attack.
- Security services: A service that enhances the security of the data processing systems using one or more security mechanisms.

Some of the most important goals of information security are listed below:

- Confidentiality: Confidentiality is the protection of transmitted data from attacks. This requires that an attacker cannot gain any useful information from analysis of the communication channels.
- Integrity: This deals with the assurance that data sent is the same as data received. No modification or removal of message data has taken place.
- Availability: An entity is not permitted to keep the computing and or communication resources from being of constructive use.
- Authentication: This assures the identities of the two communicating entities to be authentic (each is the entity it claims to be) and secondly it assures that the connection is not interfered with in such a manner that a third party can masquerade as one of the two legitimate parties for the purpose of unauthorized transmission or reception.

Possible services, mechanisms and threats are mentioned that would generally appear in most secure systems. These are explained in the ITU-T X.800 [15] and ISO 7498-2 [16] guidelines. A processing or communications service is provided by a system to give a specific kind of protection to system resources. Security policies are implemented by security services that in turn is implemented by security mechanisms. A more detailed discussion on security threats, mechanisms and services will follow later in this section.

In addition to the principles of information security there is still the question of security assurance. A sound method must be followed in formulating the security policy. This

ensures that all issues are considered and addressed. Guidelines for the management of security policies are described in the ISO 13335 [17] and ISO17799/BS7799 [18] standards. These standards describe a design method to follow to provide security assurance. There are also some organizations that audit and test the technical aspects of the system according to a set standard. An evaluation certificate is then issued stating the evaluation assurance level of the policy. A more detailed discussion on security management, risk analysis, compliance checking and assurance will follow later in the section.

2.2.1 Security Mechanisms

Security mechanisms are used to provide security services and guard against specific forms of attack [19]. Mechanisms can be divided into two classes: Specific security mechanisms and pervasive security mechanisms. Eight specific security mechanisms are defined for use in specific security services.

- Encipherment: The use of a mathematical algorithm to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and some encryption keys. This helps in authentication and key management techniques and provides both data and traffic flow confidentiality.
- Digital signature: Data appended to, or cryptographic transformation of, a data unit that allows the recipient of the data to prove the source and integrity of the data unit. It consists of a signing and a verification procedure and can be used to provide non-repudiation, origin authentication and/or integrity services and also plays a part in entity authentication.
- Access control mechanisms: A variety of mechanisms that enforces access rights to resources. These include access control lists and security labels.
- Data integrity mechanisms: Mechanisms used to assure the integrity of a data unit or stream of data units. One type can be used to provide both data origin authentication and data integrity. Mechanisms including sequence numbers and time stamps help detect replays of single data units and manipulation of a sequence of data units.
- Authentication exchanges: A mechanism intended to ensure the identity of an entity by means of information exchange. It consists of a series of cryptographic messages

exchanged between a pair of communicating entities.

- Traffic padding: The addition of data to conceal real volumes of data traffic. It is effective for providing traffic flow confidentiality if used in conjunction with encipherment.
- Routing control: Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
- Notarisation: Integrity, origin and destination of data can be guaranteed by using a 3rd party notary. It can provide non-repudiation services.

Five pervasive security mechanisms are defined which describe functions that are not related to a specific service in particular.

- Trusted functionality: That which is perceived to be correct with respect to some criteria e.g. as established by a security policy.
- Security labels: The marking bound to a resource that names or designates the security attributes of that resource.
- Event detection: Detection of security-relevant events.
- Security audit trail: Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
- Security recovery: Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

The relationship between security services and security mechanisms is shown in figure 2.1

2.2.2 Security Threats

Security threats can be divided into two categories [14]. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter resources or affect their operation. Passive attacks are difficult to detect but are easily prevented. On the other hand, it is quite difficult to prevent active attacks absolutely. Instead the goal is to detect them and to recover from any disruption or delays caused by them. Passive attacks take two forms:

Service	Encipherment	Digital Signature	Access control
Entity authentication	X	X	
Origin authentication	X	X	
Access control			X
Connection confidentiality	X		
Connectionless confidentiality	X		
Selective-field confidentiality	X		
Traffic flow confidentiality	X		
Connection integrity with recovery	X		
Connection integrity without recovery	X		
Selective field connection integrity	X		
Connectionless integrity	X	X	
Selective field connectionless integrity	X	X	
Non-repudiation of origin		X	
Non-repudiation of delivery		X	

Service	Authentication Exchange	Traffic padding	Routing control
Entity authentication	X		
Origin authentication			
Access control			
Connection confidentiality			X
Connectionless confidentiality			X
Selective-field confidentiality			
Traffic flow confidentiality		X	
Connection integrity with recovery			
Connection integrity without recovery			
Selective field connection integrity			
Connectionless integrity			
Selective field connectionless integrity			
Non-repudiation of origin			

Figure 2.1:
Service/mechanism correlation [14]

- Release of message contents (interception): An attack on confidentiality by trying to obtain the contents of a secret message.
- Traffic analysis: Opponents can determine the location and identity of communicating hosts. By observing the frequency and length of messages being exchanged the nature of communication taking place might be guessed.

Active attacks involve modification of the data stream or the creation of a false stream and can be divided into four categories:

- Masquerading: One entity pretends to be another entity in order to gain access to unauthorized resources. This is an attack on authentication.
- Replay: This involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect. This is an attack on authentication.
- Modification: A portion of a legitimate message is altered to produce an unauthorized effect.
- Denial of service: This attack prevents or inhibits the normal use or management of communication facilities. An asset of the system is destroyed or becomes unavailable or unusable.

Some threats to specific services are shown in figure 2.2. A more detailed discussion on security threats, mechanisms and services will follow later in this section.

2.2.3 Security Services

Services can be grouped into six separate categories: Authentication, access control, confidentiality, integrity, non-repudiation and availability.

2.2.3.1 Authentication

The authentication service is concerned with assuring that a communicating entity is the one it claims to be. Two specific authentication services are defined:

- Peer entity authentication: Provides for the corroboration of the identity of a peer entity in an association. It is provided during the establishment of a connection or at any other time during the data transfer phase. It attempts to provide confidence that an entity is not attempting either a masquerade or an unauthorized replay of a previous connection.

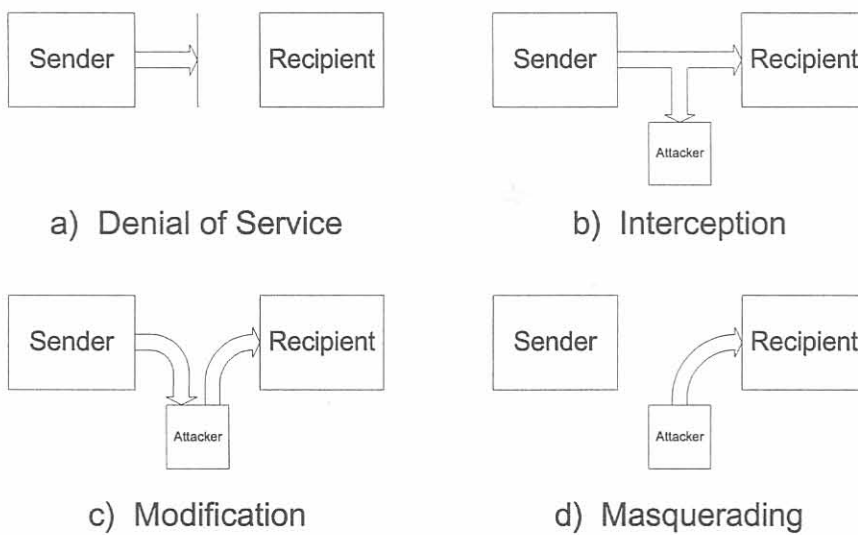


Figure 2.2:

a) Availability b) Confidentiality c) Integrity d) Authentication

- Data-origin authentication: Provides for the corroboration of the source of a data unit. Provides authentication where there were no previous interactions between communicating entities. This does not provide protection against modification or duplication attacks.

Authentication is the most important aspect to distributed system security [20]. To prevent illegal users gaining access to a node or to guard against illegal nodes or service providers gaining access to network resources, authentication is needed. This includes mechanisms to prevent replay or masquerade attacks. Users who need access to multiple computers are inevitably allocated at least one identity on each and then have the problem of remembering a mass of passwords. Therefore users prefer single sign-on systems which permit the user to authenticate themselves once and then have their confirmed identity propagated to the multiplicity of computers that are accessed. It is easy to authenticate users through the following techniques:

- Passwords
- Dynamic passwords using token devices
- Biometric authentication

Authentication measures for network entities can be classified as follows:

1. Challenge/response authentication using secret key cryptosystems:

In this measure, an authentication entity authenticates an entity by sending a challenge and receiving a response. The authenticating entity compares the response with the calculated result using the shared secret key. A three-way communication will authenticate both the sender and receiver. This requires an appropriate algorithm for strong confidentiality to be chosen, because the authentication key may be disclosed by means of attack.

2. Authentication using public key cryptosystems:

An authenticating entity authenticates by receiving the ID or the text encrypted by a sender's secret key and retrieving the ID or text by using the public key. This is similar to authentication by means of a digital signature. The only problem is that the algorithm is complex and authentication takes longer than when using secret key cryptography.

Both systems incorporate originality and integrity mechanisms such as nonces or time-stamps to prevent replay attacks. Systems incorporating time-stamps are most secure although it places stringent requirements on a synchronized timing scheme within the network. By allowing the time tolerance to be too large replays will become possible although a time tolerance that is too short might wrongly reject valid information that was delayed by network latency. In systems that lack synchronized time a three-way authentication using nonces are used [21].

Although method one is secure the scenario of each entity having a shared key is unlikely. Method two is therefore more practical as an entity's public key is easily obtained. The ITU-T X.509 recommendation is a framework where trusted certificates allow for authentication. X.509 is based on the use of public-key cryptography and digital signatures. X.509 does not specify a specific algorithm, but RSA is recommended. The digital signature scheme needs a suitable hash algorithm which will be discussed further in section 2.2.3. This system requires a trusted CA to provide certificates. The certificate is signed by the CA and can therefore be verified by anyone with the CA's public key. No entity except the CA can modify the certificate. The certificate contains the following elements:

- Version
- Serial number
- Signature algorithm identifier
- Issuer's name
- Period of validity
- Subject's name
- Subject's public-key information
- Extensions
- Signature

The ITU-T X.511 further defines tokens that is used to authenticate specific entities and exchange a secret session key once both parties obtain each other's certificates. The token is signed with the issuer's secret key. It contains a timestamp and has a short validity time. This, along with the integrity, prevents the changing of token information. The

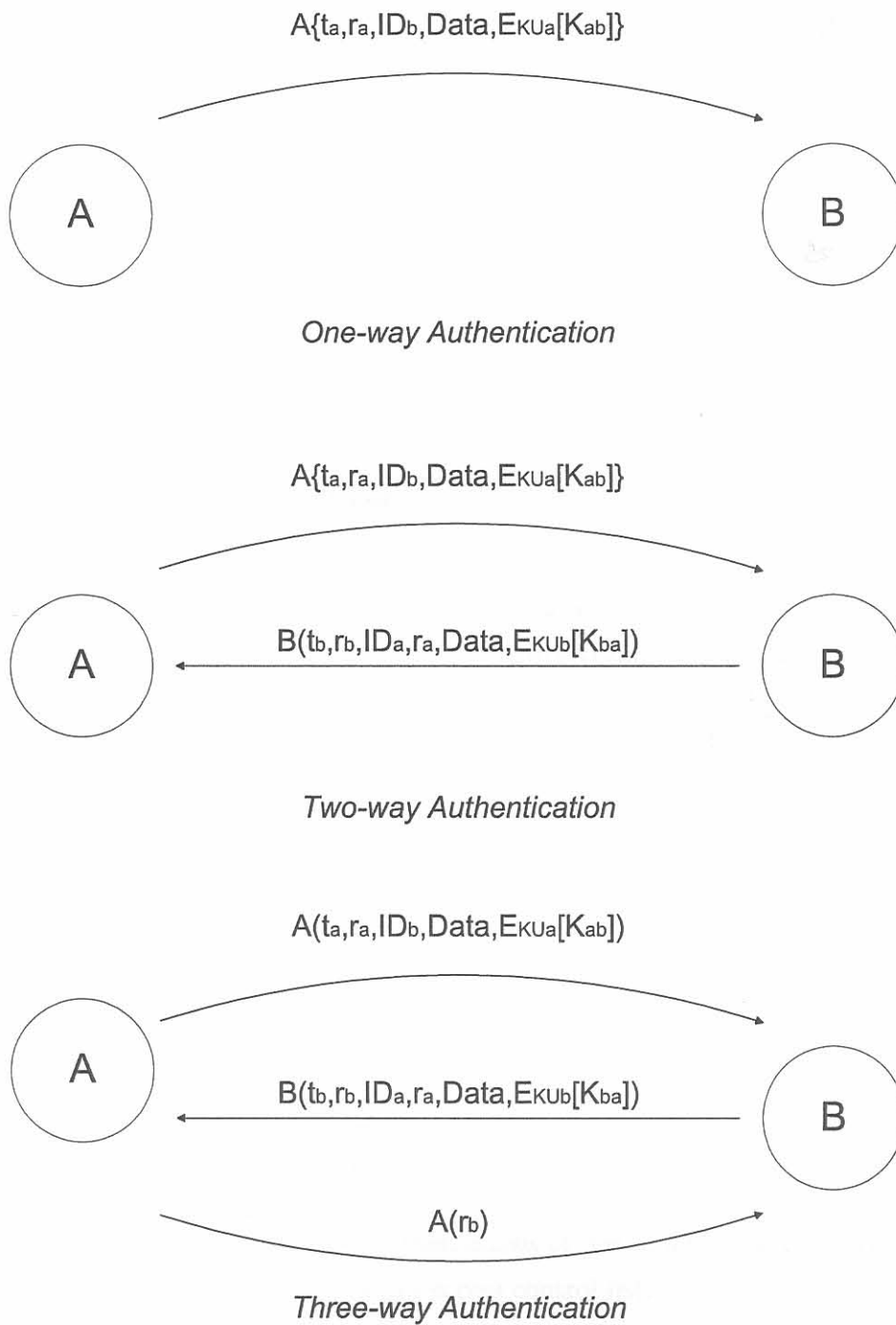


Figure 2.3:
X.509 Strong Authentication Procedures

random number and the timestamp prevent replay of the attack. The target must store all token identifiers within the validity time window in order to detect attempts of replay.

The three authentication schemes proposed by the X.509 standard using X.511 tokens are shown in figure 2.3.

- Message M encrypted by the key K is denoted $E_K[M]$
- KU_x is x 's public key.
- t_x and r_x is a timestamp and nonce from entity x
- ID_x is the identifier of entity x
- Message M signed by x is denoted $x(M)$

2.2.3.2 Access control

Access control is the ability to limit and control the access to host systems and applications via communication links. To achieve this the entity trying to gain access must first be identified, or authenticated, so that access rights can be applied to the individual. It provides protection against unauthorized use of resources and controls access to

- communication resources,
- reading, writing or deletion of an information resource, and
- execution of processing resources.

Logical access control is used to protect a computer and its information. Two sub-processes can be defined [22]:

- The user or remote system wishing access to the resources and data on a local system must be authenticated.
- The logical access control system then allows or denies access to resources based on some predefined criteria such as an access control list.

An entity must be assigned implicit or explicit rights for accessing a service. In other words the entity must be authorized to access services. Identity-based access control involves authorization criteria based on specific individual attributes. It is sometimes referred to as discretionary access control because authorization is performed at the discretion of the service owner. It is usually expressed in the form of an access control matrix.

The rows represent subjects (users, etc.) and the columns represent objects (files, service, etc.). The intersection shows the type of access the subject has to the object. In an information system with many security levels, it is not possible to enforce security with only an identity-based control policy. Discretionary controls regulate the accessing of objects, but do not control what subjects might do with the information therein. For this purpose rule-based access policies are used. These are based on a small number of general attributes or sensitivity classes that are universally enforced. Thus, all objects of the protected system must be marked with security labels. This type of access is referred to as mandatory access controls or information flow control [9].

The node used will inherit the user's security attributes in order to facilitate access control to various services [23]. Each authenticated user will have security attributes similar to these:

- Authentication level
- Group
- Role
- Confidentiality class
- Entity identity

This dissertation will not discuss in detail the rules regarding access control or any effort made to determine whether a certain user has rights to access specific services or information. It is assumed that adequate access control can be implemented if the authentication protocol is sound and secure. This includes features to prevent attacks from the public network. A firewall or access list at the gateway should be set up and resources managed in such a way that only traffic from trusted parties are permitted.

Malicious programs must be taken into consideration and prevented from entering a secure environment. A gateway with sufficient scanning software and user policies for network hosts can prevent these programs from propagating into the system. This is generally not a problem in embedded systems as software are preprogrammed in hardware and therefore malicious code cannot be added afterwards.

2.2.3.3 Confidentiality

This is the protection of transmitted data from attacks involving information leakage. With respect to the content of a data transmission, several levels of protection can be identified.

- Connection confidentiality: The protection of all user data on a connection.
- Connectionless confidentiality: The protection of all user data in a single data block.
- Selective-field confidentiality: The confidentiality of selected fields within the user data on a connection or in a single data block.
- Traffic-flow confidentiality: The protection of the information that might be derived from observation of traffic flows.

Confidentiality ensures that information is not made available or disclosed to unauthorized individuals. The only way to implement this is by cryptographic techniques. This is probably the easiest service to provide taken that implementation is basically a choice of a sufficiently strong algorithm. The choice of algorithm depends on key management, security level and available hardware. Algorithms are divided into two groups [22]:

- Asymmetric: These uses public-key cryptography such as RSA, Diffie-Hellman, Elliptic Curve.
- Symmetric: These uses shared secret keys such as DES, 3-DES, AES (Rijndael), Blowfish.

If a secure key exchange mechanism is available symmetric encryption is the better option. It is faster, in software and hardware, and therefore provides better bulk encipherment. The popular DES algorithm, 64-bit block length with 56-bit key, has been relegated to legacy systems in recent past due to security risks. The AES standard is a symmetric block cipher with a block length of 128 bits and support for key lengths of 128, 192 and 256 bits. This algorithm has clear benefits in speed and strong security and is therefore gaining in popularity.

Another important consideration in confidentiality is the placement of the encryption functions. Two options are available: link encryption and end-to-end encryption. With link encryption, each communication link is equipped with encryption devices on each end. Therefore all traffic over that communication link is secured. This is the most secure

option and also provides unrivalled traffic flow confidentiality. The disadvantage is that, because the encryption is provided in the lowest two levels of the OSI model, the data must be decrypted every time it enters a network device such as a switch or router. All the potential links in a path must incorporate link encryption and each device must share a secret key with each link partner. This involves lots of overhead and also increases the network delay. End-to-end encryption is provided in the upper layers of the OSI model and ensures that the relevant data sent is secure. This scheme requires that only the sender and final recipient share a secret key. This method is faster but offers limited protection against traffic analysis. Traffic padding mechanisms protect against traffic analysis. It is sometime possible for outsiders to draw conclusions based on the presence, absence, amount, or frequency of data exchange. Traffic padding mechanisms keep traffic approximately constant, so no one can gain information by observing it. Traffic padding is achieved by sending encrypted random data over the network.

2.2.3.4 Integrity

Integrity mechanisms provide assurance that data received are exactly as sent by an authorized entity and give protection against threats to validity of data. As with confidentiality, integrity can apply to a stream of messages, a single message, or selected fields within a message. Again, the most useful and straightforward approach is total stream protection. Five types of integrity are defined [14].

- Connection integrity with recovery: Provides for integrity of all user data on a connection within an entire data sequence, with recovery attempted. It detects any modification, insertion, deletion or replay attacks.
- Connection integrity without recovery: As above, but provides only detection without recovery.
- Selective-field connection integrity: Provides for the integrity of selected fields within the user data of a data block transferred over a connection and determines whether the selected fields have been modified, inserted, deleted or replayed.
- Connectionless integrity: Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
- Selective field connectionless integrity: Provides for the integrity of selected fields

- within a single connectionless data block and determines whether selected fields of the message have been modified.

The mechanism is used to ensure that information is not modified, inserted or removed by unauthorized individuals, entities or processes. The protection is provided through adding some authentication information to the plaintext before encryption. Another way is to use a digital signature mechanism. Digital signatures provide not only data integrity but also non-repudiation. If only integrity is required a message authentication code should suffice.

A MAC is based on applying a cryptographic hash function, $h()$ to the data that must be protected. Hash functions are much faster than conventional cryptography mechanisms. If a cryptographic hash function is applied to an input value of any length the resulting output value will always be of a constant length. The fixed length output is referred to as the message digest, checksum or hash sum. The MAC is computed in the following way:

$$MAC(M) = f(secretkey, M) = h[secretkey, h(secretkey, M)]$$

If the sender and receiver both know the secret key, the receiver can check the sender authenticity and message integrity by applying the combination of known cryptographic hash functions to the secret key and message. The most popular cryptographic hash function family is the MD family, although MD5 is specified in most documents issued by the IETF and is the latest in the family. Its 128-bit output is potentially vulnerable to birthday attacks and it is believed that it also has structural problems. SHA-1 is a better choice since it produces a 160-bit output. The input message can be up to 2^{64} bits long. The SHA-1 standard describes two methods of computation: One takes longer and uses less memory while the other executes fastest but requires more memory.

The very idea of a digital signature is that the receiver of a digital message should be able to verify the origin and integrity of the message, preferably using only public information. Digital signatures must be message dependant as well as signer dependant. There are two popular digital signature schemes based on public-key cryptography that are described by the DSS: RSA and DSA [24].

RSA requires the following public parameters [14]:

- large primes p and q

- $n = pq$
- $d \equiv e^{-1} \text{ mod } \Phi(n)$
- Private key d
- Public key (e, n)

The RSA encryption is based on the following principles:

$$\text{Ciphertext} = M^e \text{ mod } n$$

$$\text{Plaintext} = C^d \text{ mod } n = (M^e)^d \text{ mod } n = M^{ed} \text{ mod } n$$

Using RSA as the digital signature technique we need to generate S as follows:

$$S = K_{\text{Private}}(h(M))$$

The hash function is of fixed length and is usually rather short compared to the whole message. The process of generating a signature is computationally expensive so it is more efficient to use the hash than the entire message. To verify the signature it is necessary to have M , K_{Public} and the signature S . The verifier can compute the hash and compare it to the decrypted S to verify integrity:

$$\text{Is } K_{\text{Public}}(S) = h(M) ?$$

Until recently some countries prohibited the use of an algorithm in signatures that could also be used for encryption. This is the reason DSA was originally developed. DSA is based on the discrete logarithm problem and is related to the ElGamal algorithm [14]. It required much more computation than RSA in order to verify. DSA requires the following public parameters:

- large prime p
- large prime q , $q | (p - 1)$
- generator modulo p of order q , g

The signer keys consists of:

- random integer x
- $y = g^x \text{ mod } p$

The signature consists of a number pair (r, s) computed in the following way:

$$r = (g^k \bmod p) \bmod q$$

$$s = [k^{-1}(h(M) + xr)] \bmod q$$

To verify the signature the verifier computes:

$$w = s^{-1} \bmod p$$

$$u_1 = h(M)w \bmod q$$

$$u_2 = rw \bmod q$$

$$v = (g^{u_1}g^{u_2} \bmod p) \bmod q$$

Signature is valid if $v = r$

Detection of duplication replay and loss of messages is achieved with sequence numbers/time stamping before hashing. This gives the recipient the ability to verify that exchanges genuinely took place during the time interval that the time-stamp defines.

In a closed system with trusted entities integrity could be implemented using a message authentication code (MAC). This however is difficult if every node does not share keys with another and it does not provide non-repudiation. Therefore the integrity mechanism generally used in networks is built from hash functions using the RSA digital signature method.

2.2.3.5 Non-repudiation

Non-repudiation prevents either sender or receiver from denying a transmitted message.

- Non-repudiation, Origin: Proof that the message was sent by the specified party.
- Non-repudiation, Destination: Proof that the message was received by the specified party.

According to the general framework for non-repudiation as defined by ITU-T Recommendation X.813 the service comprise the following:

- Generation of proof
- Recording of proof

- Verification of proof generated
- Retrieval and re-verification of the proof

The technical means used to ensure non-repudiation services are the electronic signature of documents, the intervention of a third party witness and time stamping. Non-repudiation is also a legal concept and must also be defined by law. For public key cryptography each user is the sole owner of a secret key. Unless the whole system is compromised a given user cannot repudiate a message accompanied by his/her signature. Symmetric key mechanisms require a trusted third party witness as two entities possess the same key.

2.2.3.6 Availability

Availability can be defined as ensuring that a system's services are accessible on demand by authorized users. This service addresses the security concerns raised by denial-of-service attacks. It depends on access control and proper management and control of system resources.

Anonymous interference with communication such as denial of service must be addressed. Although there is not a specific security mechanism to provide availability it must still be considered when implementing a security policy. Sufficient authentication and access control mechanisms should prevent disruptive entities from gaining access to resources. Even then resources should be managed in such a way that no single entity can engage any resources in such a way that another entity's usage of that resource is restricted. This can be done by ensuring that entities only accept authenticated connections.

2.2.4 Security Assurance

To ensure the secure operation of a security policy infrastructure, it is necessary to have some accepted practice for the identification of security risks as well as the application of appropriate controls to manage risks. This practice is simplified by the use of formal methods and tools which increase the reliability of the system specification. Some of the relevant guideline are [25]:

- ISO/IEC 13335 or GMITS provides guidance on the management aspects of IT security.

- ISO 17799/BS7799 is a code of practice. It offers guidelines and voluntary directions for information security management. It is meant to provide a high level, general description of the areas currently considered important when initiating, implementing or maintaining information security in an organization.
- ITSEC is the existing European IT security evaluation criterion standard that allows security certification to be granted from qualifying certification bodies. ITSEC was the first example of formal security recognition between nations [26].
- COBIT provides good practices for the management of IT processes in a manageable and logical structure, meeting the multiple needs of enterprise management by bridging the gaps between business risks, technical issues, control needs and performance measurement requirements.
- Common Criteria represents the outcome of international efforts to align and develop the existing European and North American criteria towards a common standard for carrying out security evaluations [27]. By establishing a common base, the results of an IT security evaluation are more meaningful for a wider audience. CC has a catalogue of standard security functional requirements that represent the current state of the art for trusted products and systems. These can be used to develop a protection profile and as a means for developing a security target. They can also be supplemented or tailored to suit more specialist requirements. A CC evaluation is carried out against a set of predefined assurance levels, termed Evaluation Assurance Levels (EAL0 to EAL7). This scale represents ascending levels of confidence that can be placed in the TOE's security functions and determines the rigour of the evaluation:
 1. EAL1: Functionally tested
 2. EAL2: Structurally tested
 3. EAL3: Methodically tested and checked
 4. EAL4: Methodically designed, tested, and reviewed
 5. EAL5: Semi-formally designed and tested
 6. EAL6: Semi-formally verified, designed and tested

These standards provide guidelines to follow when developing a security policy. They do not provide specific solutions or suggestions on policy implementation. They rather describe a risk management methodology which ensures that a security policy is formulated

that will address the critical areas of the system being protected. Risk management is defined as the minimizing of risk by effectively applying security measures. Risk analysis is defined as a study of system assets and vulnerabilities to establish an expected loss from certain events based on estimated probabilities of the occurrence of those events. The purpose of a risk assessment is to determine if countermeasures are adequate to reduce the probability of loss or the impact of loss to an acceptable level. Figure 2.4 shows the basic risk management methodology described.

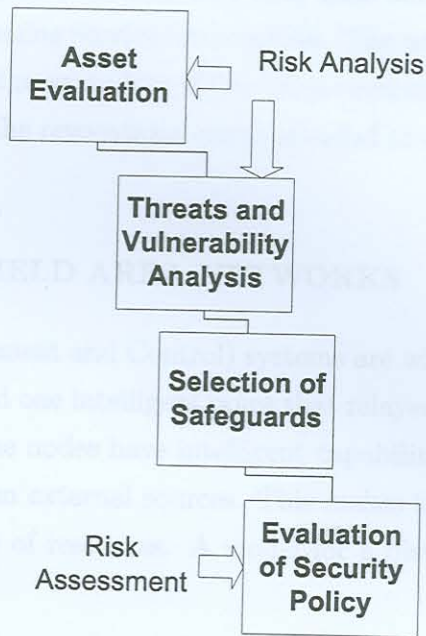


Figure 2.4:
Risk Management

2.2.5 Key Management

Key management is a process that continues through a key's entire life cycle to prevent disclosure, modifications, substitutions, reuse of expired or revoked keys and unauthorized utilizations. The secure management of cryptographic keys relates to key production, utilization, withdrawal, deletion and archiving.

The most efficient method is to have nodes share a public-private key pair. Session keys can then be exchanged at will if needed for symmetric cryptography techniques. The PKI

system also allows for easier authentication, integrity and non-repudiation mechanisms. The gateway is assumed to be trusted as it can easily be regulated that only one gateway is allowed per private network. The gateway is therefore a viable point of distribution for X.509 certificates on the private network. On the public network a trusted CA such as Thawte [28] or Verisign can be utilized. The X.509 standard also specifies the steps for creation, revocation and destruction of keys.

The risk that a key is compromised increases with time and usage. Keys have to be replaced regularly without causing service interruption. The user loses the right to a private key if the key is revealed or the secret key of the CA is compromised. All associated certificates must be revoked and the revocations communicated to all relevant verifying entities.

2.3 DISTRIBUTED FIELD AREA NETWORKS

DMC (Distributed Measurement and Control) systems are widely used in industry today. Older centralised models had one intelligent point that relayed commands to dumb nodes. In the distributed system the nodes have intelligent capabilities that allow them to function with little guidance from external sources. This makes the system more reliable and allows for more efficient use of resources. A model for a distributed system is shown in figure 2.5.

The enterprise level consists of the end user who has the ability to adjust specifications of the system. The user can also request information regarding the status of the process or the nodes. The application layer contains the protocols for the applications that is run by the nodes and relays data between the user and the node. The distributed intelligence level contains the intelligent datanodes. These nodes gain adequate knowledge from other nodes or the application layer to measure and control their respective processes. The process connection level is the way the node interacts with the process either by means of actuators or sensors. DMC systems should preferably be designed using standardised protocols that will reduce the cost of the product and promote interoperability. The nodes should have a standard interface and allow for easy maintenance and 'plug-and-play' sensors and actuators. All these aspects makes a viable DMC system with a long lifetime and maximum functionality [29]. Due to hardware considerations and the reliability required from communication networks in robust industrial systems a number of so-called fieldbus

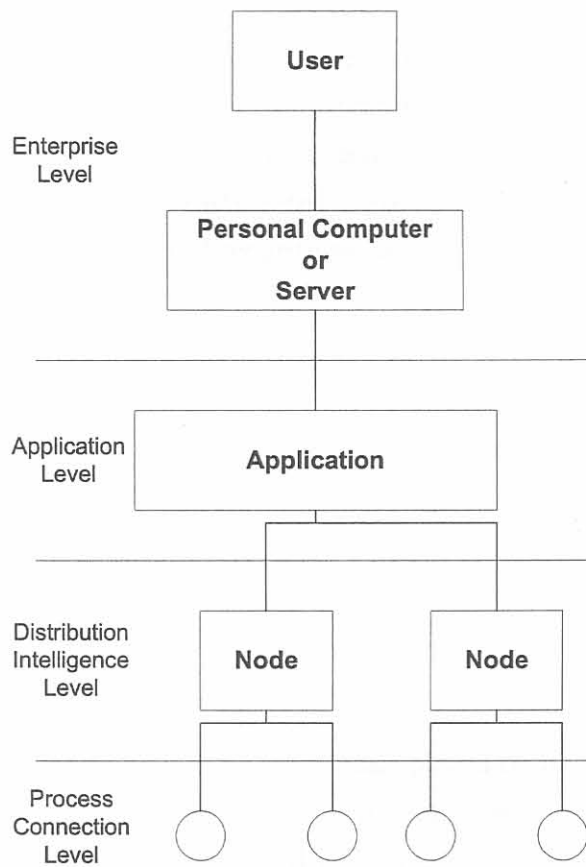


Figure 2.5:
Distributed system model [30]

protocols have been developed, such as LonWorks, Profibus, CANBus and BACNet [31]. Although not offering the connection-oriented, high-bandwidth functionality of TCP/IP Ethernet these protocols are easier to implement, offer better error-correction and are regarded as robust and dependable, which are desirable qualities in embedded industrial systems.

The low cost of computer equipment has led to a change in the way network services are delivered to end users. In recent times the computer network industry has moved away from the traditional server-client model. Information and services are no longer restricted to secure centralized control rooms and autonomous network infrastructure. Applications and information are increasingly distributed across a variety of platforms involving a mixture of vendors, technologies and security measures. This distributed network system brought many advantages [32]:

- Reliability - Accessible alternatives reduces service failure.
- Sharing of resources - Enables anyone to share or trade with everyone else.
- Aggregate computing power - Instead of one 'supercomputer' running everything the responsibilities are shared.
- Scalability - Distributed networks adapt easily to any possible number of users or services.

Real-time networks generally have time-constraints placed on its performance. Each unit of work that is to be performed is called a job and a set of related jobs that jointly provide a service is called a task. It is common to divide timing constraints into two types: hard and soft. According to commonly used definitions a timing constraint is hard if the failure to meet it is considered a fatal fault. A hard deadline is imposed on a job because a late result produced after the deadline may have disastrous consequences. In contrast the late completion of a soft deadline is undesirable but will not have a serious consequence. The requirement that all hard timing constraints must be validated invariably places many restrictions on the design and implementation of hard real-time applications as well as architectures of hardware and system software to support them. The developer of a soft real-time system is rarely required to prove rigorously that the system meets timing objectives. This allows the developer to concentrate on other performance features as well.

2.3.1 Security in Embedded Systems

Embedded systems are quickly becoming ubiquitous within our daily lives. Most people interact with an embedded system at some point during their day without knowing it. The embedded system may be in their automobile, refrigerator, or cellular phone and range from personal digital assistants to disk controllers and home thermostats to microwave regulators [4]. The key trend, however, is that all such devices are becoming more powerful, autonomous, and highly connected - following essentially the same growth curve as the Internet. The one major area in which security is addressed with regard to embedded networks is home automation. Long the futuristic domain of hobbyists, home automation is moving into the mainstream. Utility service providers offer specialized and valuable applications such as energy load management to retain customers in the face of competition. More new homes are wired for intelligent control, with security, comfort, and convenience systems becoming network aware. Homeowners can orchestrate and monitor appliances from multiple locations within the house or even remotely via telephone and the Internet, and delegate limited controls to utility service providers. By virtue of their ubiquity these systems must remain transparent to users if they are to be successful [8].

The dynamics and market forces are similar, we might suppose that the problems with security and privacy are similar as well. It would be wise to learn from past successes and failures and apply these lessons within embedded systems. Given the size and computational requirements of popular security protocols (SSL and SSH) and encryption algorithms (RSA and triple-DES), how to accomplish this task is unclear. To implement security in embedded systems is dramatically different than for full-featured, general-purpose computers. Even with today's advanced technology, embedded systems typically have severely limited resources [33]:

- Volatile and non-volatile storage are usually orders of magnitude smaller than in general-purpose computers and embedded CPU speed and available bandwidth are orders of magnitude slower.
- The capabilities of embedded systems are approximately 10 to 15 years behind the general-purpose market. Yet we still expect these systems to provide today's security levels - not those of a decade ago.

Efforts such as the Advanced Encryption Standard (AES) from the National Institute of Standards move in the right direction as the requirements for the AES algorithm included

several issues helpful to embedded systems. However, AES does not solve the entire set of problems. As a result many developers implement proprietary protocols and algorithms. This is a dangerous proposition because the approach lacks significant peer review. While it can be done, doing it right is difficult. Therefore it is required that accepted and certified technology be used. A great deal of research is still needed to provide robust security protocols supporting embedded systems [34].

Many people incorrectly view security in isolation. A single security mechanism or a single certification cannot provide adequate security. Instead, we must view security holistically, taking the overall composition of the security mechanisms and processes into consideration. This is what makes providing security an extremely difficult task [35].

It must also be accepted that no embedded hardware will ever be completely secure. IBM's 4758 is a physically secure co-processor for protecting both data and computation in potentially hostile environments. In addition to providing physical protection, its design goals encompassed the equally challenging problems of securely downloading applications into the secure environment and remotely identifying and authenticating the embedded device. The IBM 4758 was the first device to obtain a FIBS 140-1 Level 4 validation, the highest level of commercial cryptographic certification currently available. The IBM 4758 secure co-processor however has some protocol flaws. These faults make it possible to extract application secrets by following an avenue of attacks that exploit fundamental design flaws in the mathematical properties of protocol operations instead of the protocol implementation flaws that code-injection attacks exploit (e.g. buffer overflows)[25].

2.4 SMART CARD TECHNOLOGY

Smart cards in general are tamper-resistant computer microprocessor chips. They have the ability to run applications to make computations on data using programs stored in memory. This section discusses hardware architecture, software, benefits and possible applications of smart card technology.

2.4.1 Smart Card Architecture

The physical support for a smart card is a plastic rectangle on which information about the application or the issuer as well as readable information about the bearer, for example

name, date of validity, photograph, can be printed. The support can also carry either an extra magnetic strip or a bare code label. An array of eight contacts is located in accordance with an international standard. Six of these contacts are linked to the chip, which is usually not visible. They are used for power supply, ground, clock, reset and a serial data communication link [36].

A typical architecture for a smart card comprises five main components:

- The processor. This is often an 8 bit processor, the most common being Motorola's 6805 and Intel's 8048. New devices are beginning to appear in a few recent and powerful cards.
- A working memory. This is used to store temporary data when the card is in use. It is also known as the RAM.
- A ROM program memory. This contains permanent code to be executed by the processor. It should be noted that this program is stored through a mask and cannot be changed in any way.
- Non-volatile data memory. This type of memory can be written and erased over thousands of cycles.
- A communication device for exchanging data and control information between the card and the external access terminal. This communication unit works in the same way as any serial asynchronous link. The most frequent bit rate is 9600 bit/s.

For the purpose of better performance, there is often a separate cryptographic coprocessor (e.g. a modular arithmetic coprocessor for public key computations). The input/output parts and the power source differ for different types of smart cards: there are contact cards with metallic contacts, contactless cards using inductive coupling, and super smart cards with a keyboard and a display. A processor chip of a typical smart card contains three different types of memories: the working memory RAM, the maskable memory ROM, and the data storage EEPROM. The procedures and, if possible, cryptographic algorithms for general use are stored in the ROM. When an application running on an application terminal (e.g. a PC) wishes to communicate with a smart card, the card must be inserted into a card reader (also called card terminal or card accepting device).

The most important international smart card standard is the ISO/IEC 7816 [37]. This standard ensures physical compatibility between integrated circuit cards with contacts and card readers. The first specifications have focused quite naturally on the physical dimensions of the card, position of the contacts, power supply, share and duration of the electric signals, and protocols for communication between the card and the terminal. With the increase in commercial and telecommunications applications of smart cards, other parts have been added. Today's standard consists of six parts, with more to come:

- ISO 7816-1, the oldest part, specifies the physical characteristics of the card, the dimensions of the integrated circuit, the resistance to static electricity and electromagnetic radiation, the flexibility of the support, and location of the integrated circuit on the card.
- ISO 7816-2 defines the dimension and the position of the metallic contacts on the card.
- ISO 7816-3 describes the electric signals (polarity, voltage, duration, etc.), transmission protocols between the card and terminal, and the card's response to a reset originating from the terminal. Four protocols are currently defined:
 - A character-oriented half-duplex protocol identified by the value $T=0$.
 - A block-oriented half-duplex protocol identified by the value $T=1$.
 - A block-oriented full-duplex protocol identified by the value $T=2$, although this mode is rarely used.
 - The value $T=14$ indicates the use of proprietary protocols, used to support applications that preceded the standard and were already planned in France and in Germany in the health field [38].
 - The values $T=3$ to $T=13$ are reserved for future use.
- ISO 7816-4 defines the local organization of the data stored in the card and the framework for secure access to these data, in particular:
 - Cardholder authentication using a password (the PIN).
 - Authentication of an external entity using a secret key that authenticates the terminal or the bank.
 - Verification of the data integrity using a cryptogram that is often a message authentication code.

- Encryption of the data.

The ISO 7816-4 commands fall in three categories: administrative commands, security commands, and communication management commands. In general, card manufacturers prefer to pick and choose from the list of commands, so most of the cards commercially available provide only a subset of the ISO 7816-4 commands. Additional proprietary commands will be added to facilitate file and data management.

- ISO 7816-5 defines the procedure to register the application to obtain a worldwide application identifier (AID).
- ISO 7816-6 defines the inter-industry data elements.

From a technical point of view, smart cards can be classified into three main families, automata, microprocessors with simple data management and microprocessors with high-level data management. This last family is often known as multi-application cards. A common characteristic of all three families is a potentially high level of security comprising encryption using various algorithms and a distributed system of secret keys. This has been made possible by the capability of executing the required algorithms within the card itself. Such a capability does not exist with passive optical and magnetic cards [39].

The first rule of security is to gather all these five elements of a card onto a single chip. If this is not done the external wires, linking one chip to another, could represent a possible route for illegal access or use of the card. ISO standards specify the ability of a card to withstand a given set of mechanical stresses. The size of the chip is consequently limited and most of the actual constraints follow from this limitation, especially the data memory size. Chips for cards are very reliable and most manufacturers guarantee the electrical properties of their chips for ten years or more. ISO standards specify how a card must be protected against mechanical, electrical or chemical aggressions. For most existing applications a card is obsolete before it becomes damaged.

Smart card security issues can be divided into four areas:

- Card-body security.
- Hardware security.

- Operating system security.
- Card application system.

Most card-body security measures, such as embossing or hologram pictures, are designed to allow humans to check whether a card is genuine.

The smart card microcontroller (i.e. chip) must be as tamper resistant as possible. This effectively means that the cost of breaking the chip security mechanisms must be higher than the potential gain from doing so. It should be impossible to read the secret data stored on the card, such as cryptographic keys, or monitor processes running on the card and thus draw conclusions about sensitive information. Attacks against chip security can be performed at any phase of the card life cycle - card development, card manufacturing, card personalization or card use. Different attacks are performed when the chip is active (i.e. has a power supply) or inactive. Therefore it should be noted that tamper resistance does not solve all security problems and must be carefully analysed and upgraded if necessary. Security measures during card development and manufacturing include control of physical access to card data. It is also very important to implement only documented features, because undocumented features are not considered in evaluation and testing and thus can open a security hole. Each chip obtains a unique serial number, which in itself cannot protect against attacks, but serves as information for deriving cryptographic keys. During manufacture, chips are protected by authorization mechanisms based on transport codes, which can even be chip specific [36].

Most attacks on smart card hardware are performed during card use because there is practically no physical access protection. For such attacks, various rather sophisticated tools may be used, such as microscopes, laser cutters, micromanipulators, or even fast computers for probing and analysing the electrical processes on the chip. Static analysis can be made extremely difficult through special design principles such as:

- Embedded of tamper-detection mechanisms such as cover switches or motion detectors to detect, for example, cutting or drilling;
- Opaque tamper-evidence coating to hamper direct observation, probing, or manipulation of the chip surface;
- Dummy structures to confuse attackers;

- Special memory design and scrambling to hide content;
- Hiding and scrambling of buses to prevent eavesdropping.

Mechanisms that protect against dynamic analysis include:

- A voltage watchdog that switches off a chip module if the power voltage is not within a specified interval;
- Mechanisms that set to zero any parameters representing secret or private information (i.e. cryptographic keys);
- Environmental failure protection that shuts down the chip or sets sensitive parameters to zero whenever environmental conditions are outside the normal operating range (i.e. chip heating).

A dynamic attack that can determine which card command is being executed on the card (and thus potentially reveals sensitive information) is based on differential power analysis [40]. The attack works if different commands have different power consumption, so one protection mechanism is to use only commands with very similar power consumption. Another possibility is to perform the same computation (e.g. in a cryptographic algorithm) in several different ways, so that each time one way is chosen randomly. Another well-known attack is the timing attack, in which time intervals needed by the card for specific computations are measured and analysed. For example, if the card encrypts data, the greater the differences in the duration of computation for different keys and data, the easier it is to reduce the set of possible keys. A protection mechanism is to make the duration of cryptographic computations independent from input data. Attacks based on differential fault analysis try to disturb the functioning of the card (e.g. by changing the power voltage or the frequency of the external clock, or by exposing the card to different kinds of radiation). Each time the card performs symmetric or asymmetric cryptographic computations, one bit in the key is changed at some position. The result of a series of such computations, which are all different because the bit position is different in each, are analysed and used to compute the previously unknown key. The simplest protection mechanism is to let the card perform each cryptographic computation twice and to compare the results. This method is, however, rather time-consuming. A more practical approach is always to append a random number to the data to be encrypted so that attackers cannot analyse different results for the same plaintext. The random

number generator on the smart card should ideally never repeat the random numbers at any time during the card life cycle.

At present IC cards work as slaves. The program that is contained in the ROM is only an interpreter of commands coming from the outside. The protocol between the card and the co-operating device is partly standardised. It begins with a 'Reset' command that is sent to the card by the device. The response of the card is used to identify the card with respect to manufacturer information (e.g. manufacturer's code, type of card, serial production number, baud rate, type of protocol), application information (e.g. application code, security scheme) and eventually owner information (e.g. status of the card, personal security code).

There are four types of commands. The first type is used to organise the logical storage and the security scheme. For example: create or delete a logical area, give this area a name and a size, create a protection for this area or link and store a secret code. These commands are usually only available to the issuer of the card during the first session, which is called the personalisation step. The second group of commands is used by any application to manage the security scheme, i.e. to verify that any physical or logical partner that takes control of the card is actually authorised to access some of the existing areas. For example: present a secret code, present a personal identification number. The third group comprises input and output commands or, more generally, data manipulation commands: read, write, update, increase (counters for token-controlled services), decrease, compare, search, etc. The fourth group is used to add extra functions such as encryption, random-number generation, requesting unused memory size and unlocking a locked card (if that card locked itself because it has detected some attempt to bypass the security scheme). At present some vendors include some commands for non-standardised features but most of the commands ensuring basic functionality are standardized.

As a consequence, three different elements of software can be observed in a card application. The card manufacturer supplies the internal code. It is written in the enclosed microprocessor machine language. All cards of a given application contain data that are used to describe both the security scheme and the data structure. This information provides a common basis for allowing an external device to exploit all cards of the application in the same way. Every card contains specific data that characterises the owner. Finally, the external world, comprising a reader, a PC or a mainframe computer accessed through

a network, executes a set of programs to communicate with the card and manage the data held in it.

The file system defined in the standards supports two categories of files, dedicated files and elementary files. Each file has an identifier coded on 2 octets in hexadecimal notation. Figure 2.6 illustrates the way the files are arranged.

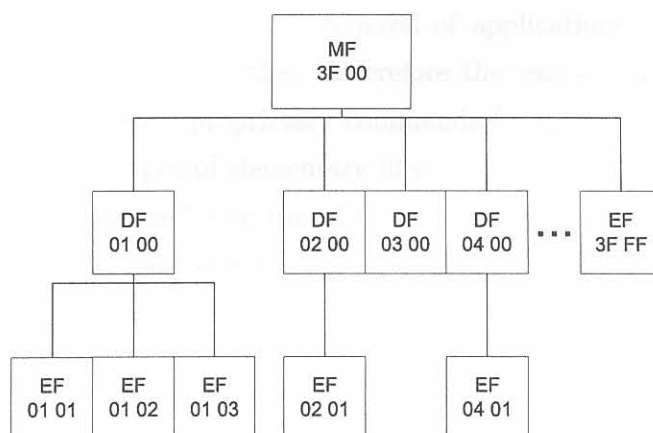


Figure 2.6:
Three structures of ISO 7816-4 file system [21]

The master file (MF) is at the root of the tree and is always identified by the file identifier 3F 00. The file identifier of the first DF is 01 00 and the last is 3E 00. Thus a card cannot contain more than 62 dedicated files in addition to the master file. Each DF is associated with a given application and may contain one or more elementary files. Application selection may be through the SELECT FILE command with the application identifier (AID) as an argument or indirectly with the help of the special elementary DIR (Directory) or ATR (Answer to Reset).

The EFs contain the data. Each EF is identified by its position in the tree, i.e., by the path leading back to the master file. The identifier is also coded on 2 octets and takes the form xx yy, where xx is the identifier of the DF to which the EF belongs. xx is 3F if the file depends directly on the master file. yy is a sequential number of the EF in that particular directory. Thus, the number of elementary files in a directory cannot exceed

63. The elementary files 2F 00 and 2F 01 under the master file have special indexing functions. The first is called DIR and the second ATR. The file DIR contain elements that allow the identification of the applications while the file ATR specifies how the card can find the application or the various objects.

The maximum number of elementary files in a card is thus $63^2 = 3969$ files. This structures is rigid and does not suit dynamic situations where files can be added or deleted corresponding to the addition or removal of applications. In fact, ISO 7816-4 does not allow the creation of new files. Therefore the various suppliers of integrated circuit cards have had to define proprietary commands for file management. ISO 7816-4 distinguishes between two types of elementary files: internal EFs and working EFs. The latter contains data for the exclusive use of entities external to the card. The internal EF contains data that the card uses during its operation. For example, in a monetary application, the following files can be present [41]:

- Key files for the storage of keys that will be used to derive a session key as specified by the payment protocol employed. Given the sensitivity of banking transactions, the applications that use purses will most probably need several keys, one for each action, such as for certification, for debit, for credit, or for electronic signature. Each key will be associated with an individual file.
- PIN files to stock the PINs that control access to the application file. The application files and the access conditions are irrevocably defined during the personalization phase.
- Purse files. For each purse, the file indicates the maximum balance, the maximum payment for each transaction, the current balance, and a backup balance to recover the previous value in case of a failure.
- Certificate files, in the case of public key encryption.
- Application usage files.

2.4.2 Smart Card Operating Systems

Development of card operating systems (COS) began in the early 1980s and today there are a dozen operating systems on the market (e.g. CardOS by Siemens, Cyberflex by Schlumberger, Multos by Maosco). COS must be kept as small (e.g. 16K) and simple as

possible in order to make testing and evaluation easy as well as to make it possible to verify whether the high security requirements are satisfied. The operating system code is written in ROM, which means that once a ROM mask has been defined and possibly millions of cards produced, no changes can be made without considerable loss of image and money. There is a range of mechanisms to make a smart card operating system as secure as possible:

- Performance of hardware, software, and memory test based on checksums at initialisation;
- Operating system design with a modular or layered structure so that error propagation is minimal;
- Hardware support to strictly separate memory regions belonging to different applications (e.g. through the addition of a memory management unit (MMU));
- Access control based on PINs.

A well-known attack is a sudden interruption of power supply, such as when a card is removed from a card reader. If performed at a precise moment, this type of attack may cause serious problems. An electronic purse may be loaded at a terminal and then removed from the reader at the very moment when the balance on the card has been increased. If the card has not yet responded to the terminal or no new audit record has been generated on the card, the terminal will believe that the load transaction was unsuccessful. The best protection against such attacks is always to use atomic transactions. This effectively means that a transaction is performed either completely or not at all. Files access control in most COS' is command based. This means that a specific command must be successfully executed before access is granted. For example, write access may be granted only after the PIN has been successfully verified by a specific command (i.e. VERIFY). An alternative is state-based access control. Basically, a state automaton is defined which specifies all allowed execution flows (i.e. command sequences) on the card. The third possibility is object-oriented access control, in which the object to be protected carries its own access control information.

There are three multi-application operating system smart card platforms (Java Card, MULTOS and Windows for Smart Card) available on the market. Commercial vendors back different technologies: VISA backs Java Card whereas arch-rival in finance, MasterCard, backs MULTOS. In the world of networked computing, Microsoft backs Windows

for Smart Card (WfSC), whereas Sun Microsystems backs Java Card. These rivals are unlikely to co-operate in developing unified standards. There are unique differences between the three MOAS platforms, each providing its own advantages and disadvantages, and each one must be considered if a new smart card system is being developed [42].

MOAS smart cards (MASCs) also need the ability to load card applications that sit side by side and provide the card holder with a variety of separate functions, such as digital identity, electronic cash, medical records and mobile phone subscriber identity module (SIM). The smart card security should be independent of the applications that are loaded to the card. MASCs are at least 20 times more expensive than an old-fashioned plastic card with only a magnetic stripe for storage. Given the cost of these highly secure devices, it makes economic sense to maximise their life span by loading and deleting card applications to meet the issuer's needs after the cards have been issued and without having to recall the cards. Therefore a secure way of dynamically controlling the card content over open networks is required. The way that MASCs are structured is as shown in figure

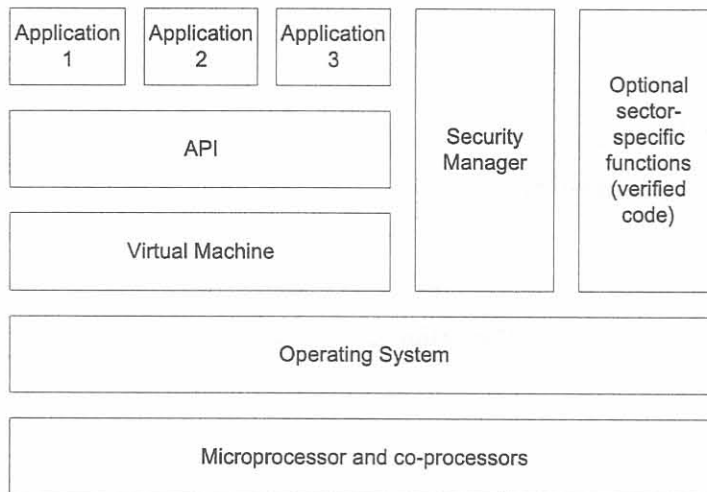


Figure 2.7:
MASC structure

2.7. The basis is a microprocessor chip implemented on a silicon chip with a certain amount of spare ROM and EEPROM. ROM masks are used during the final stages of chip fabrication to add functionality as follows:

- An operating system, allowing access in a secure and controlled manner to the raw processing power of the chip as well as useful libraries for functionality such as cryptography.
- A virtual machine that allows loaded applications to be interpreted and executed is built on top of the OS. These applications conform to the virtual machine API and are perhaps compiled from common high-level languages such as C, Java, Visual Basic, etc.
- A Card manager controlling the security of the MASC, including such functions as loading and deleting card applications and dispatching incoming commands from the terminal to the appropriate application.

It is important to an issuer to obtain supplies of devices from multiple sources and that these devices are interoperable. Therefore the adoption of unambiguous standards is paramount. A brief description of each of the standard MASC platforms that are currently being marketed is given in the following sections. Further details of how the platforms compare are given in table 2.4.2.

2.4.2.1 Java Card overview

Java Card was launched in 1995-96, the first major release was in 1997, and the current release is Java Card 2.1.1 [43]. The specifications are in three parts:

- API: the programmer interface.
- Virtual machine: providing binary portability between Java Cards.
- Runtime environment.

The Java Card Forum [44] develops and recommends specifications to Sun Micro-systems Inc., owner of the Java Card VM and API specifications. Java Cards have had their most success in the mobile telecommunications sector where standards are emerging for SIMs and SIM toolkits. The Java Card standard does not specify an operating system. It also does not specify how Java Cards should dynamically manage their applications. In this

Table 2.4.2:
Summary of MASC products

	Java Card	MULTOS	WfSC	Open Platform
Specification Control	Sun Microsystems	MAOSCO	Microsoft	Global Platform
Current Versions	Java Card 2.1.1	MULTOS 4	WfSC1 .1	OP 2.0
Certification	Unproven	ITSEC E6 - Highest level possible	Unproven	Not applicable
Scope	Java API access run-time environment through virtual machine	Complete OS: Virtual Machine, API, run-time environment, management	Multi-functional, cannot really support many applications	Application management
Portability	Virtual machine byte-code portable, unproven	Proven portability	Not yet	Unproven
Personalization	Personalization after loading application	After or before loading application	Non yet	Only provides life-cycle management
CA Structure	Cost not known	Cost: US\$ 0.04 per application reload	Cost not known	Not applicable
Benefits	Popular Java development tools	Interoperable, secure, stable and efficient	Flexible, familiar Microsoft tools	Global PIN and secure communication channels
Disadvantages	Security, stability, interoperability not proven	Inflexible, no cheaper non-RSA option	Security concerns	Not proven. Bugs in cards and compilers

respect, most Java Cards are migrating towards using the emerging Open Platform standard. Some Java Cards, claiming to be JC 2.1.1 compliant and OP 2.0 compliant, are beginning to emerge. They have yet to demonstrate true interoperability where applications can be compiled once and then loaded onto several Java Card implementations and be guaranteed to execute correctly. Java Card interoperability is further hampered

by the fact that no independent security evaluation is required. All major smart card manufacturers have Java Card platform offerings. Some Java Card licensees claim that they will be targeting Common Criteria EAL4+ and 5+, but it will be the best part of a year before this might be achieved. Java Card applications are developed in Java using standard tools that developers may well already have. However, it is not yet possible to buy a source-level debugger for Java Card applets.

2.4.2.2 Open platform overview

VISA developed the VISA Open Platform [45] specifications to try to fill some of the interoperability gaps in the Java Card specifications. Actually the VOP specifications go further than this in that they also address terminal and personalisation issues. In October 1999, VISA handed over the Open Platform specifications to the Global Platform consortium [46] to manage and encourage the adoption of OP outside financial circles. The OP specifications address the issue of how to manage the card applications. A scheme is proposed that addresses areas such as:

- Dynamic secure loading and deletion of applications
- Secure communication channels between the issuer and the card or card applications
- Global card PIN which can be shared by applications

As with earlier Java Card specifications, emerging platforms using VOP or OP may suffer from incomplete specification and not be able to interoperate. For example, initialisation of the global PIN is specified by VOP but not by OP. The area of most concern is that the control of secure dynamic application management is not specified in complete detail. Implementations of OP will inevitably be incompatible, which will be a disadvantage as far as card issuers are concerned. Open Platform is not limited to Java Cards. In the future we may see OP cards with underlying WfSC technology. In the long term, this may allow some degree of card management interoperability between platforms using OP.

2.4.2.3 Windows for smart card overview

Microsoft's core business is the sale of operating systems. As such they will not allow anyone else to implement Windows for Smart Card operating systems [47]. Microsoft wants to control new technologies (mobile phones, digital TV, palm devices and access to networks) by ensuring the ubiquity of their operating systems. They realise that the smart card will be key to all of these. WfSC has some interesting properties that set

it apart from the other two open standard MASC platforms. One of these is that the issuer can configure the card operating systems and decide which parts should be present on their cards. The intention is that optional modules will include functionality such as GSM and cryptography support, and support for industry standards such as ISO7816-4 [37]. The application developers decide which of these are needed for their applications and the cards can be manufactured accordingly. This leads to the possibility of extremely cheap low-functionality, multiple sources of supply and third-party security evaluation. The \$2 cards that Microsoft promised have yet to appear and are unlikely to if they are to have a useful amount of application memory on them. WfSC does not impose an issuer-centric security model. If the system requires a card where the cardholder controls what applications are on this card, WfSC is the most likely platform. This is much more like the PC model, but has the same security implications.

Since the announcement of WfSC in 1998, Microsoft has repeatedly failed to deliver on its promises. While WfSC has come a long way from the first developer release in May 1999, it is still to be proven to work reliably and be mature. The current version WfSC v1.1 is more of a multi-function smart card than a true multi-application smart card. It does not fully conform to the industry standard for communicating with smart cards at the application level, ISO 78164. While WfSC is not yet a mature commercial product, what is certain is that, with Microsoft's resources, they will have a credible offering eventually. The power behind the name of Microsoft has already proven enough to make large international organisations align with WfSC. Microsoft is also a tool provider. By making familiar tools available at give-away prices, they are guaranteeing that many developers will experiment with creating applications for their MASC. Only the Microsoft Visual Basic compiler is available at present, which produces highly inefficient code, though better ones are likely to appear as WfSC becomes mature. In common with Java Card, WfSC does not specify how the card content is managed securely post issuance. Microsoft spokesmen say that they will wait to see what the market requires. Open Platform is likely to be an option.

2.4.2.4 MULTOS overview

MULTOS has merged from the banking sector [48]. It was developed by the NatWest Development Team in the UK as a secure platform for an electronic purse. It is the only non-military product in the world to achieve ITSEC E6 High certification, which is the highest certification available [26]. ITSEC is the European predecessor of the emerging Common Criteria for Information Security Evaluation. A consortium called MAOSCO controls the MULTOS specifications. As MOASCO believes that security is important, it insists that any implementation of MULTOS is evaluated and achieves certification to ITSEC E6 (High). This is done at the cost of the implementer and takes typically over a year to achieve.

MULTOS is a genuinely interoperable unambiguous standard. MULTOS has included everything required of a MASC (the OS, virtual machine and card manager). If you buy a MULTOS card you know exactly how it will work and switching between suppliers is done with minimal effort. The MULTOS specifications are mature and have been stable for around three years. The MULTOS virtual machine (or Application Abstract Machine, as they call it) is tailored for smart cards and as such allows very efficient (in both speed and size) applications to be written. In order to develop for MULTOS, compilers for C or Java are required, or machine-code-like native MEL may be used for ultimate efficiency (though this is rarely necessary). Only one good compiler toolset is available for MULTOS. It is not possible to ask for a MULTOS card without the secure dynamic application management facility (which uses RSA cryptography), or without the ability to perform cryptography for digital IDs. Thus MULTOS requires a co-processor for RSA and therefore uses relatively expensive underlying chips. However, most smart card applications anyway require RSA (digital ID, Windows 2000 logon, EMV, CEPS e-purses). The secure application management uses digital certificates, which at present can only be obtained from the MULTOS CA based in the UK. This model may be unattractive to potential issuers and MAOSCO plans to offer to license the CA to third parties or issuers themselves. MAOSCO is relatively open about its plans for future generations of the MULTOS standard. Published in late September 2000, MULTOS 5 adds various features including an optional dual interface (contact and contactless) to a single MULTOS chip, and support for GSM. Perhaps with the announcement of mobEcom's SecureSim on MULTOS, there will be a serious challenge to Java Card's dominance of the GSM market.

2.4.2.5 Comparison

The smallest code is that written in MEL for MULTOS. This is not surprising since the MULTOS virtual machine is optimised for smart card processors. Code generated from applications written in Java for Java Cards or C for MULTOS cards are around the same size, though usually the MULTOS is slightly smaller still. The code generated by the Microsoft VB compiler is much larger than for equivalent applications written for other MASCs. This seems to be largely due to compiler inefficiency and is likely to be fixed in the future.

WfSC execution speeds are not considered since the platform is not yet stable enough. The MULTOS code in MEL is fastest of the three platforms. MULTOS applications coded in C average 25% slower unless there are a lot of primitive calls, in which case there is no noticeable difference in speed. Java Card applications have been anything up to 50% slower than the equivalent MULTOS applications, though it has to be said that there are still many problems getting general applications to compile, load and execute on a range of Java Cards. It is interesting to note that Java Card SIMs are implemented as verified code rather than Java Card applets due to the performance issues with Java Card. In contrast, the SecureSIM on MULTOS from mobEcom is implemented in C as a MULTOS application that is executed in the virtual machine.

Each of the three MASC platforms has unique selling points. If you need interoperability with multiple sources of supply of known security level and with the ability to dynamically load and delete applications post issuance then MULTOS is the only choice. Java Cards compliant to JC 2.1.1 and OP 2.0 may emerge and begin to successfully interoperate over the next year or so. The remaining issue will still be their level of security. WfSC is not a mature offering at this point in time. Microsoft has the resources to move quickly and so WfSC may well soon be a serious contender. The three platforms will become more and more difficult to differentiate. For the time being, it is likely that the three will continue to operate in their own niches: Java Card largely in the mobile SIM world, MULTOS in long-term, high-volume rollouts where stability, known security and post issuance download are required and one day WfSC for use in applications in Windows environments [42].

2.4.3 Why Use Smart Cards?

The need to manage and secure a rapidly growing information network has focused increasing attention on smart card technology. Over the past decade, smart cards evolved from offering basic memory to complex systems with chips that incorporate powerful processing units with dedicated peripherals. This evolution enabled a wide range of applications. Smart card applications include financial transactions, e-commerce, physical access control, health and transportation services, and access to such wireless systems as the global system for mobile communication (GSM) and the upcoming universal mobile telecommunications system (UMTS) third-generation mobile phones [6].

Such applications depend on smart cards equipped to perform onboard cryptographic digital-signature encryption and authentication. Smart card operating systems use these cryptographic features to manage data storage and control access to private information. Essentially smart cards serve as security tokens by securely storing users' personal data and service providers' private information. The card interacts within a system using special communication interfaces and dedicated protocols. Smart cards provide highly reliable mechanisms for storing, accessing, and using data in non-volatile memory. Data access control and data management follow a security policy based on cryptographic service and defined for a specific application.

Sensitive data such as personal information, secret keys, and private application information stored in smart card memory is protected by combined hardware and software mechanisms. The write/store operation is more aggressively protected in smart cards than in any other device. Special onboard security sensors prevent alterations to memory during data storage or reading. In addition, the software includes a backup mechanism in case of card power-down during storage. Access and storage mechanisms can be combined of typical OS access management and systematic cryptographic verification to authenticate the application or to ensure transaction confidentiality. Smart card hardware features like a hardwired firewall between memory areas can make storage even more secure. Even if chemical or electrical corruption alters memory, hardware memory integrity checks or a software checksum will detect the alteration. Finally, hardware and software protect against illegal reading of smart card data. These combined measures offer powerful assurance of data privacy. Access management is far more secure in smart cards than in any computer OS. Objects, files, and keys can be protected during read, write, and execution

by secret codes or authentication-granted access rights set up by keys used in symmetric cryptography, such as DES, or asymmetric cryptography, such as RSA algorithms.

To ensure authentication, confidentiality, and integrity through cryptography, smart cards have enhanced arithmetic computation capabilities. Typical smart cards use secret-key algorithms such as the well-known Data Encryption Standard, the Advanced Encryption Standard, or other proprietary algorithms specified by operators. These algorithms mainly use data substitutions, permutations, compression and table lookups, and Boolean-to-arithmetic conversions. These generally simple operations lead to fast implementations even when performed in a high-level language. The associated keys are short (from 56 to 256 bits) and quite easy to manage. High-end smart cards offer far more powerful cryptographic algorithms, known as public-key algorithms. Examples include RSA for encryption/decryption and digital signatures. These schemes require an arithmetic unit to compute modular multiplication and reduction on large numbers, because the keys are at least 512 bits long and may reach 2,048 bits. With either type of algorithm, chips may have dedicated peripherals such as DES or RSA cryptoprocessors for efficiency. The OS would use such peripherals, for example, during the authentication scheme, either directly or by adding software to enhance hardware security [36].

Today's cards contain at least 128 Kbytes of ROM, associated with 64 to 128 Kbytes of EEPROM or flash memory, and 4 to 8 Kbytes of RAM. This compares with 16 Kbytes of ROM, 4 Kbytes of EEPROM, and 256 bytes for RAM offered a few years ago. Silicon technology enabled most of this progress by reducing the transistor scale for smart cards. To overcome external clock limitations, chips now run with their own internal clock, independent or not of the external one. Chips were using slow external clocks, even for heavy internal computations such as public-key cryptography [49]. Existing smart card terminals were not able to provide higher frequencies and modifying all the terminals was impractical. Chips with asynchronous communications provided the solution: The CPU runs its own clock and uses an external clock only for communication. CPUs and their peripherals can now run a 30-MHz internal clock, increasing chip speed by a factor of 6 to 10.

Chapter 3

SYSTEM OVERVIEW

3.1 SYSTEM DEFINITION

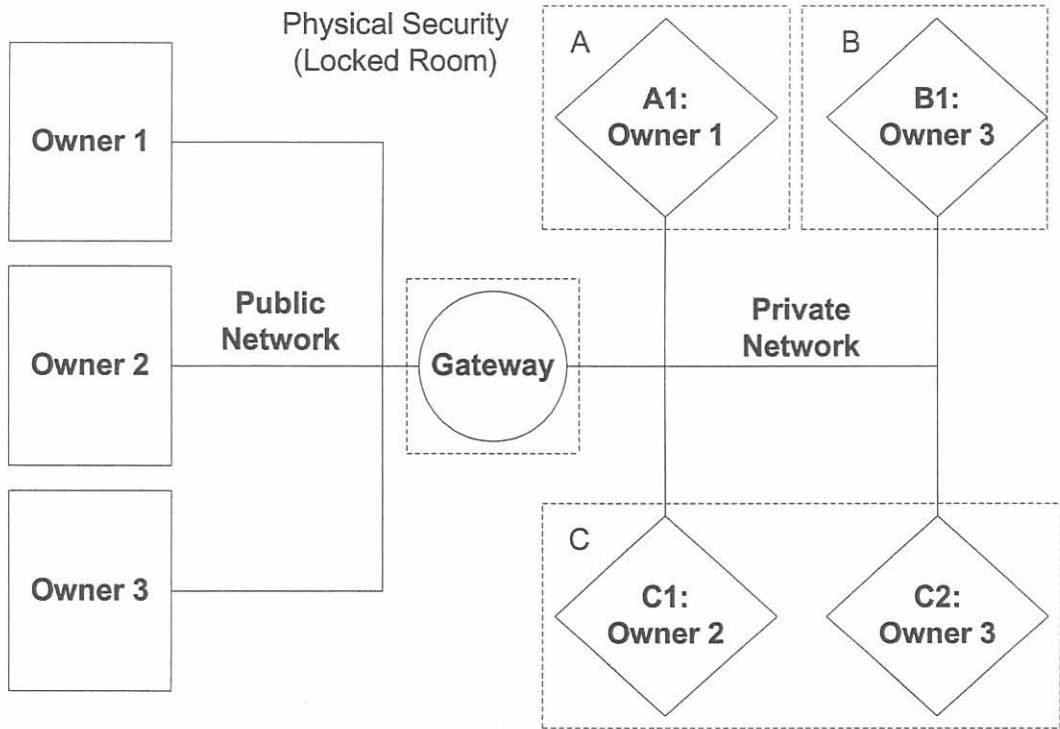


Figure 3.1:
System Model

The general configuration for a possible field area network, connected to a public network, is shown in figure 3.1. The public network allows for different entities to access the private network from remote locations. The private, or field area network, is responsible for connecting the entities located at the users' premises. This dissertation will refer to entities on the private network as nodes, to the entities in the public network as owners (owners can utilise the node to gain information or to provide a service) and to the entity linking the two networks together as the gateway. Such a system is extremely versatile, it accommodates various setups and both the user and the owner can make requests for a service or data. This can be applied in multimedia systems, prepaid systems and commercial applications (e.g. the user could request a service online which then causes its node to be enabled) [8]. The private network is not autonomous. Entities from different owners or

entities with different users may be connected to the same private network. Each entity needs a secure way to relate to its owner, user and peers in the network without being compromised [2].

3.1.1 System Architecture

The following system entities can be identified (please refer to figure 3.1):

- Premises A, B and C. These are areas controlled by different users, e.g. a single room, an area of a factory and a single house or flat.
- Embedded nodes A1, B1, C1 and C2. Located at premise A, B and C respectively, these nodes provide service to the local user.
- The private network (FAN) provides a communication medium between the nodes and the gateway.
- The gateway connects the private network to the public network. It is unlikely that the private and public networks utilize the same protocols and therefore the gateway can also be seen as protocol translator.
- Owners 1, 2 and 3. These entities access the nodes they own in order to provide services to the users.
- The public network allowing remote access to the gateway by the service providers.
- The user who benefits from the action of a node.

3.1.2 Assumptions

In order to focus on the problem area the boundaries of the system need to be defined. In order to limit the system certain assumptions need to be clarified:

- Premises:
 1. The premises are physically secure.
 2. The nodes cannot be physically accessed by anyone except the user.
 3. A premise might contain nodes related to different service providers.

- Nodes:
 1. Nodes on the same private network can communicate with one another.
 2. Nodes communicate with the gateway to facilitate communication with its owner.
 3. A node only communicates with its owner to receive instructions and relay information.
 4. The node's owner and user might not be the same entity.
 5. A node only has one owner.
 6. A node only has one user.
 7. A node is physically secure.
 8. A node has low processing and storage resources.
 9. A node's software cannot be altered after implementation.
 10. The node is limited in what security services it can provide.
 11. A node is not a trusted entity.
 12. The frequency of communications between nodes can be high and consists of short data streams.
- Field Area Network:
 1. The implemented private network infrastructure spans multiple premises.
 2. A central IT management structure controls possible addressing schemes and the gateway.
 3. Low-bandwidth, connectionless message based communication, e.g. CANBus, Profibus, LonWorks.
 4. Communication standard only specifies layer 1 and 2 of the OSI model.
 5. Broadcast or bus architecture, which means all entities on the network receive all data transmitted.
- Gateway:
 1. There is only one gateway in the system for each field area network.
 2. The gateway is a trusted entity if adequate physical security is ensured.

3. The gateway is untrusted when adequate physical security cannot be ensured.
 4. The gateway has sufficient resources to implement any security service.
 5. The gateway only relays data or executes commands from a service provider.
 6. The gateway needs only to communicate with owners that own nodes on the private network.
- Owner:
 1. Communicates with the nodes on different private networks.
 2. Communicates with different gateways.
 3. The owner is a trusted entity.
 4. The frequency of communications between an owner and a node is low and consists of long data streams.
 - Public Network:
 1. Public infrastructure, e.g. Internet.
 2. High bandwidth communication, connection oriented communication, e.g. probably TCP/IP.
 3. The operation of this infrastructure is ensured by a third party and is not the responsibility of either the user or the owner.

3.2 RISK ANALYSIS

In order to perform a risk analysis the assets and the basic functional requirement of the system must first be identified. Adequate safeguards must be implemented to protect the assets and ensure that the functional requirements are met. Vulnerabilities and threats need to be identified in order to determine which safeguards our policy must implement [50].

3.2.1 Identification of Assets

Information is the most important as the whole system is implemented in order to share information between entities for the purpose of making correct decisions in providing functionality. If data was to be corrupted or prevented from reaching its intended recipient the network would not be able to function correctly. Therefore all assets are rated

according to their ability to maintain successful data flow. The system assets, in order of importance, are as follows:

1. Nodes: Allow users to access the network and obtain several services. If the users cannot gain access to the network the system is no longer functional as it has no purpose. Nodes also store and manipulate data to provide useful information. If a node is prevented from functioning correctly all functionality is lost.
2. Private network infrastructure: Allows the nodes to communicate with another in order to provide functionality within a limited geographic area. Disruption of the private network is disastrous as nodes cannot obtain information or supply information that might be critical to the overall function of the network, e.g. a central air conditioner might require temperature readings from various sources in order to make adjustments. Functions that rely on a single node will still have limited functionality, e.g. a valve regulating a cooling tank might keep the water level correct but will not be able to accept changes or report water level readings to its owner.
3. The gateway: Although not providing any services the gateway is solely responsible for all traffic flow between the networks. Failure in the gateway will therefore seriously limit connectivity and data flow outside the private network. The gateway is also centrally located which makes it a viable centre of operations for security services.
4. Public network infrastructure: This network allows the owner to access its nodes and provide a service. Due to the nature of distributed embedded systems (the functionality is provided locally as far as possible) the private area network and its applications will still function. However, the owner won't be able to access the node in order to retrieve information or make changes to its services.
5. The owners: Although actions by the owners are not as critical as the actions made by the nodes these entities still provide and require services from their nodes that are necessary for overall system functionality.

3.2.2 System Requirements

The following functional requirements necessary for system operation can be identified:

- F1: Two nodes must be able to communicate with one another.

- F2: A node must be able to communicate with the gateway.
- F3: An owner must be able to access its node when required.
- F4: The gateway, a unauthorized node or a unauthorized entity on the public network must not be able to gain useful information from a communication sequence between a node and another authorized node, or a node and its owner.
- F5: A node needs to be authenticated by other nodes and its owner.
- F6: An owner needs to be authenticated by its nodes and the gateway.
- F7 A untrusted gateway must be authenticated by the owner.
- F8: The gateway must relay valid messages between the private and public network without being able to disclose information within those messages.
- F9: An owner, node or gateway may gain access to any system resources illegally.
- F10: Entities must be assured of the integrity of any information transmitted.
- F11: Information and services are private and may not be disclosed.
- F12: Necessary proof must be provided in order to settle disputes between parties.

3.2.3 Vulnerabilities and Threats

Vulnerabilities are aspects of the system which can be exploited. Threats can be seen as possible ways in which these vulnerabilities can be exploited to undermine the functional requirements [51]. The areas of vulnerability, with associated threats, can be listed as follows:

1. Node

- N1: Denial of service attack.
- N2: Use of modified data or data from a malicious source.
- N3: Release of message contents between node and another authorized entity.
- N4: Authentication of a malicious owner (masquerade, replay attacks).
- N5: Authentication of malicious user.

- N6: Unauthorized owner being allowed to access services to which he/she is not entitled.
- N7: Unauthorized user being allowed to access services to which he/she is not entitled.
- N8: User or owner of node denying that service was received or requested.

2. Private Network (FAN)

- FAN1: Node or malicious entity gaining access to a network which it is not entitled to.
- FAN2: Illegal node pretending to be another legal node.
- FAN3: Modification of data sent or received.
- FAN4: Release of message contents between entities on the network.
- FAN5: Authentication of a malicious node on the Private Network (masquerade, replay attacks).

3. Public Network

- PN1: Denial of service attack on the public network infrastructure cannot be addressed.
- PN2: Modification of data sent or received.
- PN3: Release of message contents between owner and private network.
- PN4: Traffic analysis on public network.

4. Gateway

- G1: Denial of service attack (private/public network side).
- G2: Node gaining illegal access to public network and owners.
- G3: Outside entity (or owner) gaining illegal access to private network and nodes.
- G4: Authentication of a malicious user or node on the private network (masquerade, replay attacks).
- G5: Authentication of a malicious owner on the public network (masquerade, replay attacks).

- G6: Attack of the gateway in order to gain access to owner-to-node communication.

5. Owner

- O1: Denial of service attack
- O2: Authentication of a malicious node/gateway (masquerade, replay attacks).
- O3: Illegal node gaining access to service to which it is not entitled.
- O4: Illegal gateway gaining access to services to which its network is not entitled.
- O5: Release of message contents between owner and gateway/node.
- O6: Modification of data between owner and gateway/node.
- O7: User node denying that service was received or requested.

Only vulnerabilities of technical nature will be considered, therefore threats as a result of personnel security, business management or training inefficiency are ignored. Table 3.2.3 shows the final risk analysis of the system. Likelihood of occurrence is mapped versus severity if the threat should realise. Overall importance is numbered from 1 to 9, with 1 being the most important.

Table 3.2.3:
Risk analysis for distributed system

	Frequent	Not often	Seldom
High	1 N1, G1	2 N2, FAN6, PN1, G3, O1	3 N4, N5, N8, G6
Medium	4 FAN1, FAN2, PN2	5 N7, FAN5, PN3, G4, G5, O5	6 N6, O4, O7
Low	7 FAN3, FAN4, O2, O3	8 N3, G2, O6	9 PN4

Chapter 4

Chapter 4

SYSTEM SPECIFICATIONS

4.1 SECURITY POLICY OVERVIEW

Many systems that need security are concerned one way or another with monitoring some aspect of the environment. They range from ordinary domestic burglar alarms through utility meters to tachographs, and even a number of systems critically concerned with nuclear safety. The protection of these systems is most often more concerned with preventing attacks that involve denial of service, such as swamping communications, overwhelming sensors. Service denial attack might be complemented with various kinds of data manipulations. Systems may have to deal with numerous mutually suspicious parties, and often are implemented using the cheapest possible microcontrollers. Many of these systems are continually in the hands of the enemy. Key management is an important consideration, especially in low-cost widely distributed systems where a central key management facility cannot be justified. The services required are determined from the functional requirements to be met and risks that must be addressed. The possible security services are:

- S1: Access Control
- S2: Authentication
- S3: Confidentiality
- S4: Integrity
- S5: Non-repudiation
- S6: Availability

Table 4.2 shows the various functional requirements and the security service that will be needed to make ensure that each requirement is met.

4.2 SYSTEM COMPONENTS

In order for a security policy to be successfully implemented each component of the system must address certain security aspects. In section 3.2.1 the system's main components were identified. Threats to each of these components, with regards the functional requirements, were listed in section 3.2.3. This section explains where security services must be implemented in order to mitigate threats to the system assets and functional requirements.

Table 4.2:
Functional Requirement vs Service mapping

	S1	S2	S3	S4	S5	S6
F1						X
F2						X
F3						X
F4			X			
F5		X				
F6		X				
F7		X				
F8	X	X				
F9	X	X				
F10				X		
F11			X			
F12					X	

4.2.1 Node

In order to facilitate security services the node needs to have the ability to perform cryptographic operations, e.g. encrypt/decrypt, hash and generate random numbers. The node also needs to store some information pertaining to the respective security services: data about some nodes, the gateway, its owner and its user. To prevent replay attacks the node must keep track of communication sequence numbers and time-stamps (needs a clock). Mapping of applicable threats to security services are shown in table 4.2.1.

- N1: Denial of service attacks at the node will mostly be of a physical nature. Nodes must be physically secure to prevent access to security and functional hardware.
- N2: If the node accepts and processes malicious information it will be prevented from providing a valid service. If it is controlling a critical operation the consequences will be disastrous. Nodes must be able to verify the integrity of all information received.
- N3: Information release does not effect the operation of the node but does effect its owner. A malicious entity could intercept sensitive information about business oper-

ations and sell it to competitors. The node must secure all information transmitted or received.

- N4: If a malicious owner convinces a node that it is legal then that owner can disrupt the system or gain sensitive information. Authentication procedures must be secure and ensure that only legal entities are authenticated.
- N5: If a malicious user convinces a node that it is legal then that user can gain access to services. Authentication procedures must be secure and establish the identity of the user correctly.
- N6: A legal owner might try to access information that it is not allowed to see. Information must be labelled and access control implemented at the node in order to determine what the owner may access.
- N7: A legal user might try to access information that it is not allowed to see. Information must be labelled and access control implemented at the node in order to determine what the user may access.
- N8: If the user or the owner denies that a transaction or service took place there must be sufficient proof to settle the dispute. The owner and the user must be successfully identified and some secret information from both must be used to bind both entities to a transaction.

Table 4.2.1:
Threats: Node vs Service mapping

	S1	S2	S3	S4	S5	S6
N1						X
N2		X		X		
N3			X			
N4		X				
N5		X				
N6	X	X				
N7	X	X				
N8					X	

4.2.2 Private Network

In order to facilitate security services the network needs to provide some information: destination ID, source ID, acknowledge flag, end of message flag, multicast flag and sequence number. Most fieldbus protocols only define the operation of the lower 2 layers of the OSI model. Therefore a network and transport layer must be implemented to provide an addressing scheme and connection-oriented communication. Mapping of applicable threats to security services are shown in table 4.2.2.

- FAN1: If a malicious node gains access to the network and convinces other entities that it is legal then that node can disrupt the system or gain sensitive information. Procedures must be implemented to ensure that only valid nodes reside on the network.
- FAN2: If a malicious node successfully appears to be another legal node then that node can gain access to services. Authentication procedures must be secure and establish the identity of the node correctly.
- FAN3: If a node or gateway accepts and processes malicious information it will be prevented from providing a valid service. All entities must be able to verify the integrity of all information received.
- FAN4: Information release does not effect the operation of the network but does have consequences. A malicious entity could intercept sensitive information about network operations and use it in future attacks. All information transmitted or received must be secured by network entities.
- FAN5: If a malicious node gains access to the network and convinces other entities that it is legal then that node can disrupt the system or gain sensitive information. Authentication procedures must be secure and ensure that only legal nodes can reside on the network.

Table 4.2.2:
Threats: Private Network vs Service mapping

	S1	S2	S3	S4	S5	S6
FAN1	X	X				
FAN2		X		X		
FAN3				X		
FAN4			X			
FAN5		X				

4.2.3 Public Network

The protocols in the public network are well known and provide services for the 3rd and 4th layer of the OSI model already. Therefore existing functionality can be used. Mapping of applicable threats to security services are shown in table 4.2.3.

- PN1: Denial of service attacks on the network infrastructure cannot be addressed by the security policy. The only solution is to choose a reliable third party network operator with a sound security policy pertaining to availability.
- PN2: If a gateway or owner accepts and processes malicious information it will be prevented from providing a valid service. All entities must be able to verify the integrity of all information received
- PN3: Information release does not effect the operation of the network but does have consequences. A malicious entity could intercept sensitive information about network operations and use it in future attacks. All information transmitted or received must be secured by the communicating entities.
- PN4: Although no sensitive data is released an attacker can still gain information about network topology or the nature or frequency of transactions from control information, e.g. addresses, time-stamps, etc. The communication must be secured in a way such that some control information is hidden but the public network can still process the packets.

Table 4.2.3:
Threats: Public Network vs Service mapping

	S1	S2	S3	S4	S5	S6
PN1						X
PN2				X		
PN3			X			
PN4			X			

4.2.4 Gateway

In order to facilitate security services the node need to have the ability to perform cryptographic operations, e.g. encrypt/decrypt, hash and generate random numbers. The nodes also needs to store some information pertaining to the respective security services: data about the nodes on its network and their owner. To prevent replay attacks the node must keep track of communication sequence numbers and time-stamps (needs a clock). Mapping of applicable threats to security services are shown in table 4.2.4.

- G1: Denial of service attacks at the gateway can be physical or originate from either the private or public network. The gateway should be physically secure if possible. Resource sharing must be managed in such a way that it impossible for one entity to tie up the gateway. The gateway should only accept service requests from authenticated entities that are less likely to disrupt services.
- G2: A legal node might try to request information from an owner that it is not allowed to see. Information must be labelled and access control implemented at the gateway in order to determine what services the node may request.
- G3: A legal owner might try to access nodes which it is not allowed to. Nodes must be labelled and access control implemented at the gateway in order to determine what the owner may access.
- G4: If a malicious node convinces the gateway that its legal then that node can gain access to services. Authentication procedures must be secure and ensure that only legal nodes can reside on the network.

- G5: If a malicious owner convinces the gateway that it is legal then that owner can gain access to the private network. Authentication procedures must be secure and ensure that only a legal owner can access the private network.
- G6: The gateway has access to all owner-node communication. A physical or software attack might try to gain access to the gateway in order to monitor, modify or disrupt communications. The gateway must be physically secure and a trusted administrator must ensure its operation. Unauthorized software installations or data access must not be allowed.

Table 4.2.4:
Threats: Gateway vs Service mapping

	S1	S2	S3	S4	S5	S6
G1						X
G2	X	X				
G3	X	X				
G4		X				
G5		X				
G6	X		X	X		

4.2.5 Owner

In order to facilitate security services the owner needs to have the ability to perform cryptographic operations, e.g. encrypt/decrypt, hash and generate random numbers. The owner also needs to store some information pertaining to the respective security services: data about its nodes and the gateway. To prevent replay attacks the owner must keep track of communication sequence numbers and time-stamps (needs a clock). Mapping of applicable threats to security services are shown in table 4.2.5.

- O1: Denial of service attacks at the gateway can be physical or originate from the public network. The owner IT infrastructure is governed by its security policy. The owner is responsible for ensuring availability.
- O2: If a malicious entity convinces an owner that it is legal then that entity can disrupt the system or gain sensitive information from the owner. Authentication procedures must be secure and ensure that only legal entities are authenticated.

- O3: A legal node might try to request services that it is not allowed to. Services must be labelled and access control implemented by the owner in order to determine what the node may access.
- O4: A legal gateway might try to request information on behalf of a network which is not entitled to that information. Access control must be implemented by the owner in order to determine if the gateway may request a service, e.g. labeling the data with security levels and preventing entities to access data with higher clearance.
- O5: Attackers will attempt to access sensitive information communicated between the owner and its nodes. Information transmitted or received must be secured.
- O6: If the owner accepts and processes malicious information it will be prevented from providing a valid service. The owner must be able to verify the integrity of all information received.
- O7: If the node or the owner denies that a transaction or service took place there must be sufficient proof to settle the dispute. The owner and the user must be successfully identified and some secret information from both must be used to bind both entities to a transaction.

Table 4.2.5:
Threats: Owner vs Service mapping

	S1	S2	S3	S4	S5	S6
O1						X
O2		X				
O3	X	X				
O4	X	X				
O5			X			
O6				X		
O7					X	

Chapter 5

IMPLEMENTATION

This chapter describes the implementation of a security policy that will address the threats identified by the risk analysis in section 3.2. It describes in detail how the security services identified in chapter 4 can be implemented using smart card technologies. It further elaborates on the technical requirements and specific implementation for each system component. A policy overview is given that describes the security protocols used to provide the security services. Finally possible implementations of mechanisms are given that could be used to implement the protocols.

5.1 SECURITY SERVICES

5.1.1 Availability

In this system the greatest risk is denial of service attacks. These attacks could occur frequently and require little technical expertise or knowledge of the particular network system e.g. flooding of network with useless messages, physical attack on infrastructure. These attacks are easily detected but difficult to prevent. Preventing attacks from the public network could be accomplished by a firewall implementation at the gateway. Owners should protect their systems in a similar way. Public traffic will only be allowed once a secure association has been made between the owner and the gateway or node. The possibility of a denial of service attack originating from the private network is low but is still a possibility. Nodes must only accept connections from authenticated entities, a message received from an unauthorized entity should be disregarded and trigger an alarm. The network management must be alerted immediately if any suspect behaviour is observed by a node or a gateway.

5.1.2 Authentication

Authentication is the second largest concern in the system. Users, hosts or service providers and even server nodes might pretend to be a legal entity in order to gain access to the system with malicious intent. The first step is to determine which entities must be authenticated, and by whom. In order to meet the system's functional requirements the following need to take place:

- user authentication by the node
- node authentication by other nodes, the gateway and its owner
- gateway authentication by the owner and the gateway

- owner authentication by the gateway and its nodes

The authentication protocols can be based on public-key or private key cryptography and are shown in section 5.4. The necessary key pairs are generated when a new user, node, gateway or owner is registered. Adequate security checks (e.g. management authorization, background check) must be performed to ensure that these entities are valid. These key data must be stored in a physically secure location and kept for a mandatory length of time in order to provide information for the settling of disputes. Each server, host and gateway also maintain a revocation list of key pairs that have been compromised before their expiry time. This prevents compromised keys being used to gain access to the network.

In the network any user can gain access to services from a node. Taken that a node only has one user it is possible for that node to store unique information that can identify the user. Depending on hardware requirements the user can be authenticated using a PIN or biometric information, e.g. fingerprint profile can be stored on the node's smart card [20]. This legally binds the user to the node.

To prevent illegal entities from accessing the network all nodes on the private network must authenticate themselves before communication between the two entities commences. The gateway keeps record of all nodes housed on the private network. The broadcast nature of the network protocols ensures that the gateway can monitor all network traffic. If a unregistered node transmits on the network the gateway will attempt to authenticate this node. If authentication fails an alarm will be raised. If a node sees that it receives network traffic from a node using its identity it must also raise an alarm. Symmetric or asymmetric authentication mechanisms may be used.

5.1.3 Access Control

Users or nodes gaining access to resources that they are not entitled to are mostly due to problems with authentication. User A might masquerade as User B in order to get B's privileges. A secure authentication protocol should prevent this. The second problem is to allow an authenticated entity access to only specific services, and to regulated what that entity is entitled to do. The simpler the access control conditions the easier it is to implement. An owner can only be accessed by a node it owns while a node can only be accessed by its owner or another node sharing the private network. Nodes and

information on these nodes must be given security labels which indicate clearance levels similar to multi-level security systems like La Padula and Biba [52],[53]. Node-to-node access control can then be done on a node basis (each node has a clearance level, to access a node you need equal or higher clearance) or on a data basis (data has a clearance level, to access data you need equal or higher clearance). Some other examples of access control rules are given below:

- Write/Read: Can the user only read data or also modify data ?
- Owners: Specific assets might only be available to certain owners.
- Clearance: Specific information can only be viewed by users with adequate clearance.

When a node is registered it is given a fixed clearance which cannot be altered by anyone except the owner. Registration is the process of generating a smart card (e.g. adding keys, biometric information and access control information), inserting it into the node (physically sealing it) and connecting the node to the network for the first time (node registers with gateway and owner online before starting to operate).

5.1.4 Integrity and Non-repudiation

Digital signatures provide both integrity and non-repudiation when used in conjunction with a hash function. The hash function which is recommended is SHA-1 although smart cards also provide the MD5, RIPEMD and DES based algorithms. Smart cards provide DSA and RSA PKI signature techniques but private key MACs can also be implemented. A MAC does not offer the same level as DSA or RSA signatures because either of the recipient or sender could have created the signature. RSA has the added benefit of facilitating key exchange. DSA is slightly faster but due to the nature of the algorithm secret information cannot be transferred. It does however prove that a transaction took place between two entities. After the authentication protocol is completed the two communicating entities should have enough information to verify each other's signatures. Two verification processes must be completed:

- The plaintext user information is signed by the providing entity. This provides non-repudiation and ensures integrity of information. This also pads the message sufficiently to ensure higher security for the next signature.

- The encrypted data transmitted on the communication line is signed by the sender. This protects against modification during transmission. Integrity can therefore be checked without decrypting the information.

All entities are required to keep the necessary information to settle disputes. A record of transactions, with signature and required key pair must be kept. This record must be backed up and stored for an indefinite period. The laws governing the specific provision of service might specify such a mandatory time period.

5.1.5 Confidentiality

The authentication protocol implemented must allow for the exchange of a session key. This session key along with a suitable encryption algorithm should prevent leakage of content. Due to the processing and storage constraints it would be more efficient and even more secure to use symmetric key algorithms. It is recommended that the DES algorithm be implemented, although smart cards also provide 3-DES and IDEA, while AES should be provided in the near future. In the distributed system given in section 3.1 there is limited value to traffic flow confidentiality on the private network. In this case it is more important to ensure that the user or service data is not leaked. Traffic confidentiality is still a concern on the public network. Traffic padding is to be used to prevent traffic analysis on the network. Due to the fact that traffic padding might consume unnecessary resources it can be applied only to critical applications. Due to the end-to-end nature of the system only the payload and communication fields not used to route messages (e.g. sequence numbers) may be encrypted.

5.1.6 Key Management

Key management is a process that continues through a key's entire life cycle to prevent disclosure, modifications, substitutions, reuse of expired or revoked keys and unauthorized utilizations. The secure management of cryptographic keys relates to key production, utilization, withdrawal, deletion and archiving. Before the implementation of the above services can be implemented in detail at the system component level it is crucial that it is determined what roles public and private key structure will play in the security policy.

The risk that a key is compromised increases with time and usage. Keys have to be replaced regularly without causing service interruption. The most difficult aspect of this system is the distribution of shared keys between entities to facilitate data confidentiality.

The most efficient method is to have entities share a public-private key pair. The PKI keys are assigned to the node with its implementation and never need to be changed. If a secret key is compromised the node gets assigned a new identity. Session keys can then be exchanged at will if needed for symmetric cryptography techniques using the X.511 authentication sequences. The PKI system also allows for easier authentication, integrity and non-repudiation mechanisms. The system's storage constraints make it impractical for each entity to store another entity's public key as inexpensive smart cards usually have only 4K memory and cheap microcontrollers have even less. Taking into account that these keys vary from 512 - 1024 bits it is feasible that a smart card can store its owner's public key and its own public-private key pair. The X.511 authentication methods will also take considerable time and processing effort. Symmetric secret keys are only 64-bits long for most implemented algorithms, so more of them can be stored. Therefore the node could store the most recent keys used in a similar way to a cache system in a PC. A node can also store a master symmetric key. This key can be used to derive different session and authentication keys. The problem is that the entity it communicates with must also have this master key. Therefore another technique is still needed to distribute master keys.

The TTP CA model of public key distribution is also impractical because the node cannot afford to store another public key, even if it is temporary. If the gateway is a trusted entity it could distribute keys between the nodes. If the gateway cannot be trusted the key distribution might have to occur between the owners. In some cases nodes can be preconfigured with shared master keys if it is known that they will communicate with one another on the private network. The master key never leaves its secure storage and if a key is compromised another can be derived. Although this is feasible if both nodes belong to the same vendors or if the system implementer has the rights to program the security features it does not allow for ad hoc connections between nodes or later network changes. The system requires both asymmetric and symmetric key management. Owners and gateways have greater resources and therefore they can use the protocols described in X.509 and X.511 standards. These entities are also connected over a public network. This allows for authentication using public key certificates provided by a trusted CA. How the keys are managed by the different components will be described further in section 5.2.

5.2 SYSTEM COMPONENTS

In order for a security policy to be successfully implemented each component of the system must address certain security aspects. In section 3.2.1 the system's main components were identified. Threats to each of these components, with regards the functional requirements, were listed in section 3.2.3. This section explains how security services are implemented in order to mitigate threats to the system assets and functional requirements.

5.2.1 Node

The smart card at the node supplies random number generation, symmetric encryption/decryption, hash functions and PKI. This allows the node to secure its data and implement the necessary security service. The nodes provides the following:

1. Time.
2. Sequence numbers of message communicated with different entities.
3. Support for operational commands. The six basic operations are:
 - The gateway must send out broadcast commands regularly to determine whether all nodes are still functioning. Node must respond with a signed identity. This also provides traffic padding on the private network.
 - Node must send data when needed.
 - Node must update its data when needed. This includes receiving and executing instructions, or updating security information.
 - Node must be able to initiate or respond to an authentication request.
 - Node must implement the communication protocol used by the private network.
 - Node must raise an alarm if it detects security violations or suspicious behaviour, e.g. receiving a message using this node's ID as the source ID.
4. Data about the service it is monitoring or providing:
 - Data identifier.
 - Data descriptor.
 - Data format, e.g. ASCII, INTEGER, FLOAT.
 - Read/Write access.

- Length.
 - Security level of data. Any entity that wished to access this data must have adequate clearance.
5. Information about transactions and actions. Owner will retrieve and store this records to perform a security audit.

The smart card stores data about a number of entities:

1. User, this information authenticates the user:
 - PIN number used to identify the user.
 - Biometric information used to identify the user. This requires newer smart cards with 16-32K of memory.
2. Node
 - The identity of the node.
 - Security level of the node. Used by other nodes to implement access control.
 - The public-private key pair and/or master key used for authentication, non-repudiation and integrity.
3. Owner
 - The public key of the owner. Used in authentication between node and owner. When using a master key with symmetric cryptography this is not needed.
 - The owner's identity.
 - The symmetric key used to communicate with its owner. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.
4. Other nodes
 - The symmetric keys used to communicate with the other nodes. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.
 - The identities of the nodes.

5. Gateway

- The public key of the gateway. Used in authentication between node and gateway. When using a master key with symmetric cryptography this is not needed.
- The gateway's identity.
- The symmetric keys used to communicate with the gateway. Assures the confidentiality of transmitted data. This key is timestamped when it is exchanged and expires after a set time.

One concept needs to be emphasized: Each node has one smart card acting as a security application module. The node and the user inherit the security attributes of that smart card. Therefore the smart card determines the user and the node's access control clearance. It also provides all the mechanisms and information for authentication, integrity, confidentiality and non-repudiation. If a message from a non-authenticated entity is received or the integrity of the message fails the message is immediately discarded. This should prevent unauthorized entities from tying up the node's resources and provide availability.

5.2.2 Private Network

The network itself does not provide much security services but a protocol must be specified to provide functionality for the network entities. Most fieldbus systems only implement the physical (1st) and data link (2nd) layers of the OSI model. Therefore the system designer must implement the network (3rd) and transport (4th) layers. In each case it must be stated whether data can be protected (P) (confidentiality and integrity) or whether it is send unprotected (U). The following must be provided for each data frame:

1. Multicast Flag (U): Used to indicate a broadcast message.
2. Destination identifier (U): Identifies the intended recipient of the message.
3. Acknowledge flag (U): Used to indicate an acknowledge message.
4. Error Flag (U): Indicates that an error has occurred. Used in an acknowledge message.
5. Last message flag (U): Indicates the end of a data sequence.

6. Source identifier (U): Identifies the sender of the message.
7. Sequence number (P): Used to prevent replay attacks.
8. Instruction identifier (P) : Instruction identifier.
9. Data (P): The payload.

The destination and source identifiers are also appended to the payload (once per sequence) and compared to the advertised values once the message is recovered and verified. The sequence number, the acknowledge flag and the error flag provide connection-oriented communication. The acknowledge message will contain the sequence numbers of all the messages it is acknowledging. The gateway must send out broadcast commands regularly to determine whether all nodes are still functioning. Nodes must respond with a signed identity to prove they are still functional. This also provides traffic padding on the private network. All the security services needed are provided by the network entities (the gateway and nodes) and not by the actual infrastructure.

5.2.3 Public Network

The protocols in the public network are well known (TCP/IP) and provide services for the 3rd and 4th layer of the OSI model already. Therefore existing functionality can be used. To provide confidentiality traffic padding can be implemented. The owner can regularly poll the gateway or some of its nodes to see if they are still active. These entities can respond with a signed identity. The TTP CA is also housed on the public network although the security concerns for a CA is beyond the scope of this dissertation.

5.2.4 Gateway

The gateway has sufficient resources to implement random number generation, symmetric encryption/decryption, hash functions and PKI algorithms. It also has unlimited storage and processing resources compared to the node. This allows the gateway to secure its data and implement the necessary security service. The gateway provides the following:

1. Time.
2. Sequence numbers of messages communicated with different entities.
3. Support for operational commands. The five basic operations are:

- Gateway must relay node-to-node and node-to-owner communication.
 - Gateway must be able to initiate or respond to an authentication request.
 - Gateway must implement the communication protocol used by the private network.
 - Gateway must implement the communication protocol used by the public network.
 - Gateway must raise an alarm if it detects security violations or suspicious behaviour, e.g. receiving an alarm from a node.
4. Information about the nodes on the private network:
- Node identifier.
 - Node descriptor.
 - Information on node's owner.
 - Node's public key or master key.
 - Symmetric key used to transmit secure data to that node. This key is timestamped when it is exchanged and expires after a set time.
5. Information about the owners of nodes on the private network:
- Owner identifier.
 - Owner descriptor.
 - List of nodes owned by the owner.
 - Owner's public key and certificate.
 - Symmetric key used to transmit secure data to the owner. This key is timestamped when it is exchanged and expires after a set time.
6. The public-private key pair used for authentication, non-repudiation and integrity.
7. Information about the operation of the private network. This entails sufficient information for a security audit.

The gateway can either be a trusted entity or it can be assumed that it is untrusted. An untrusted gateway is still authenticated by the nodes and the owners so it not an unauthorized entity. A gateway is seen as untrusted if it can be physically attacked or

the possibility exists that malicious entities can gain access to the communications on the gateway. A gateway secures its communications with both the nodes and the owners. All message are signed to ensure integrity and non-repudiation. A firewall might be implemented at the gateway to provide additional availability and access control services to the private network and the gateway. The gateway is the main access control point in the system. The owners are authenticated and prevented from accessing nodes they do not own. Nodes are authenticated to ensure that only valid nodes reside on the private network.

5.2.5 Owner

The owner has sufficient resources to implement random number generation, symmetric encryption/decryption, hash functions and PKI algorithms. It also has unlimited storage and processing resources compared to the nodes. This allows the owner to secure its data and implement the necessary security service. The owner provides the following:

1. Time
2. Sequence numbers of message communicated with different entities.
3. Support for operational commands. The four basic operations are:
 - Owner must be able to communicate with its nodes, gateways and other owners.
 - Owner must be able to initiate or respond to an authentication request.
 - Owner must implement the communication protocol used by the public network.
 - Owner must raise an alarm if it detects security violations or suspicious behaviour, e.g. receiving an alarm from a node.
4. Information about the nodes on the private network:
 - Node identifier.
 - Node descriptor.
 - Node's public key or master keys for itself and the gateway.
 - Symmetric key used to transmit secure data to that node. This key is time-stamped when it is exchanged and expires after a set time.
 - Transaction data from the node used for security audit.

5. Information about other owners:

- Owner identifier.
- Owner descriptor.
- List of nodes owned by that owner.
- That owner's public key and certificate.
- Symmetric key used to transmit secure data to that owner. This key is timestamped when it is exchanged and expires after a set time.

6. Information about gateways:

- Gateway identifier.
- Gateway descriptor.
- List of nodes owned by the owner on that private network.
- Gateway's public key and certificate.
- Symmetric key used to transmit secure data to the owner. This key is timestamped when it is exchanged and expires after a set time.

7. The public-private key pair used for authentication, non-repudiation and integrity.

The owner needs to communicate with its nodes to provide services and obtain information. Sometimes other owners will need their nodes to speak to a node it owns and therefore ask for permission. All gateways, owners and nodes are authenticated before transactions can take place. An owner secures its communications with the nodes, gateways and other owners. All messages are signed to ensure integrity and non-repudiation. A firewall might be implemented at the gateway to provide additional availability and access control services to the owner's IT infrastructure.

5.3 SECURITY POLICY OVERVIEW

The overall system security policy comprises of different protocols that are used under different circumstances.

5.3.1 Node Registration

This protocol describes the steps taken when a node is placed on a private network for the first time. These steps provide the basis for later security operations. Please refer to figure 5.1. Two different protocols are proposed: Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.1.1 Methods

Scheme 1:

1. User A supplies some secret information, e.g. password or biometric data. User A is authenticated by Node A
2. Node A send its ID, its access rights and its owner's ID to the gateway in plaintext. The message is signed by Node A.
3. Gateway authenticates the owner:
 - a) Gateway requests owner's certificate from CA. If gateway still holds a valid certificate for the owner this step is skipped.
 - b) Owner requests gateway's certificate from CA. If owner still holds a valid certificate for the gateway this step is skipped.
 - c) Gateway authenticates owner and exchanges session key K_{GO} .
4. Gateway sends the node ID to owner. Communication is encrypted using K_{GO} and signed by the gateway.
5. Owner records Node A's location and the gateway's information (e.g. address, public key and certificate).
6. Owner checks its records and sends the gateway Node A's public key. Communication is encrypted using K_{GO} and signed by the owner.
7. Using Node A's public key the gateway verifies Node A's signature of the information send in step 2.
8. Gateway send its public key to Node A.
9. Gateway authenticates Node A. Session key K_{GA} is exchanged.

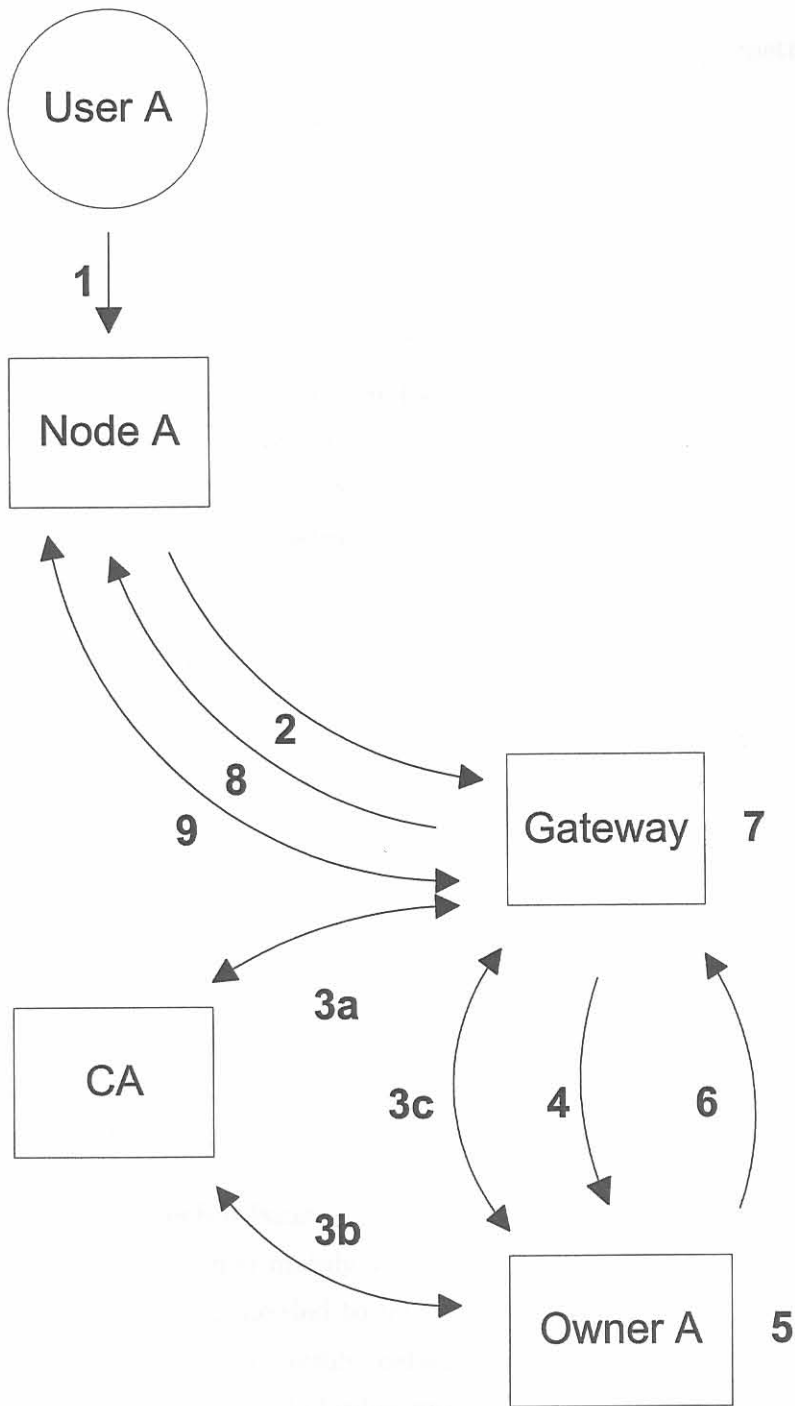


Figure 5.1:
Security protocol for node registration

Scheme 2:

1. User A supplies some secret information, e.g. password or biometric data. User A is authenticated by Node A
2. Node A send its ID, its access rights and its owner's ID to the gateway in plaintext. The message is signed by Node A using master key K_{MGA} or a derived key,.
3. Gateway authenticates the owner:
 - a) Gateway requests owner's certificate from CA. If gateway still holds a valid certificate for the owner this step is skipped.
 - b) Owner requests gateway's certificate from CA. If owner still holds a valid certificate for the gateway this step is skipped.
 - c) Gateway authenticates owner and exchanges session key K_{GO} .
4. Gateway sends the node ID to owner. Communication is encrypted using K_{GO} and signed by the gateway.
5. Owner records Node A's location and the gateway's information (e.g. address, public key and certificate).
6. Owner checks its records and sends the gateway Node A's master key, K_{MGA} . Communication is encrypted using K_{GO} and signed by the owner.
7. Using Node A's master key, K_{MGA} or a derived key, the gateway verifies Node A's signature of the information send in step 2.
8. Gateway authenticates Node A using an exchange protected by the master key or by a derived key. Session key K_{GA} is exchanged or is derived from the master key.

5.3.1.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.

- X.511 authentication procedure recommended.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication and key exchange sequence on the private network.
4. Public key cryptography must provide digital signature and key exchange on the public network.
 - RSA only available algorithm to provide both.
 - X.511 authentication procedure recommended.

5.3.2 Owner \Rightarrow Node

This protocol describes the steps taken when an owner initiates communication with a node. Please refer to figure 5.2. All communications between an owner and a node must be secured in such a way that end-to-end security is provided. The gateway must not be able to decipher the messages even if it is trusted. Two different protocols are proposed: Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.2.1 Methods

Scheme 1:

1. Node A authenticates the gateway and exchanges a session key K_{GA} . If a previous session key is still valid this step is skipped.
2. Node A exchanges a session key K_{OA} with its owner. If a previous session key is still valid this step is skipped.
 - a) Node and owner know each other's public keys.
 - b) Node requests an owner authentication from the gateway.
 - c) Gateway and owner mutually authenticate and exchange session key K_{GO} . If a previous session key is still valid this step is skipped.
 - d) Gateway forwards the request to the owner.
 - e) Gateway relays authentication sequence between owner and node.

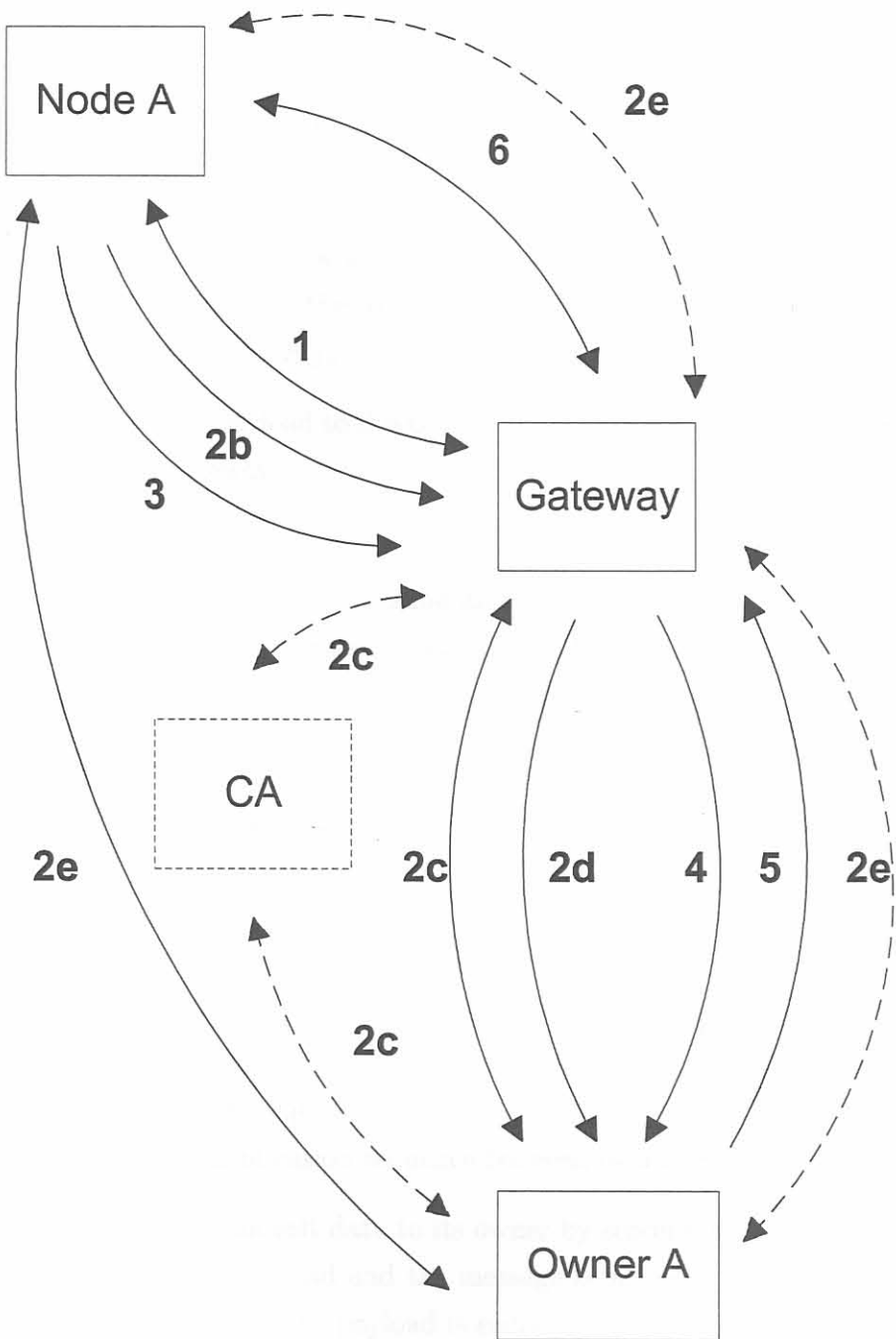


Figure 5.2:
Security protocol for owner ⇒ node communication

3. The node requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
4. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.
5. Response is generated by the owner and forwarded to the gateway. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
6. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.

Scheme 2:

1. Node A authenticates the gateway using an exchange protected by the master key, K_{MGA} , or by a derived key. Session key K_{GA} is exchanged or is derived from the master key, K_{MGA} . If a previous session key is still valid this step is skipped.
2. Node A authenticates owner using an exchange protected by the master key, K_{MOA} , or by a derived key. Session key K_{OA} is exchanged or is derived from the master key, K_{MOA} . If a previous session key is still valid this step is skipped.
 - a) Node and owner know the master key K_{MOA} .
 - b) Node requests an owner authentication from the gateway.
 - c) Gateway and owner mutually authenticate and exchange session key K_{GO} . If a previous session key is still valid this step is skipped.
 - d) Gateway forwards the request to the owner.
 - e) Gateway relays authentication sequence between owner and node.
3. The node requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
4. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.
5. Response is generated by the owner and forwarded to the gateway. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .

6. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.

5.3.2.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.
 - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
 - Non-repudiation is not needed between the gateway on the node.
 - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.3 Node \Rightarrow Owner

This protocol describes the steps taken when a node initiates communication with its owner. Please refer to figure 5.3. Two different protocols are proposed. Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.5.3.1
Scheme

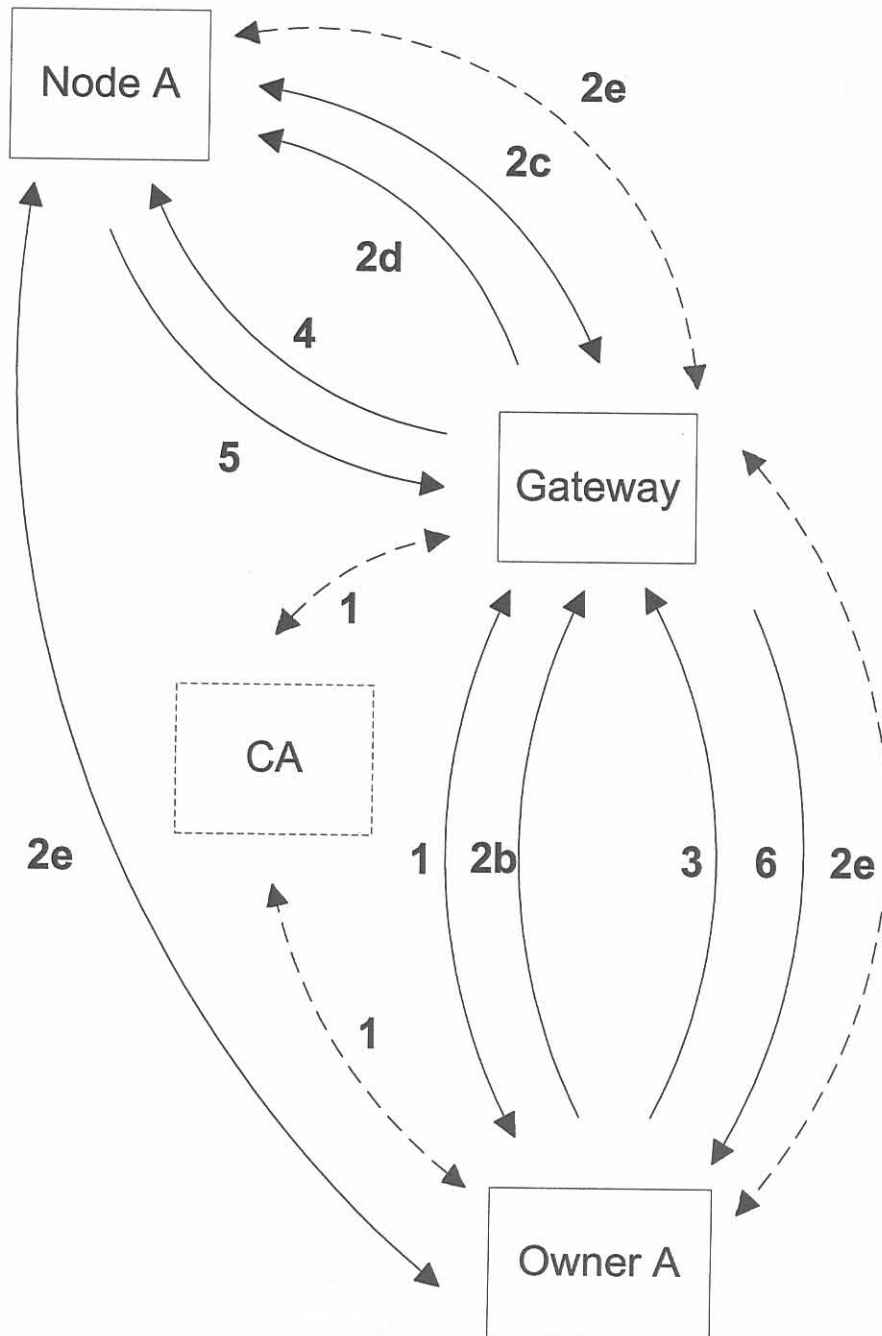


Figure 5.3:
Security protocol for node ⇒ owner communication

5.3.3.1 Methods

Scheme 1:

1. Owner authenticates the gateway and exchanges a session key K_{OA} . If a previous session key is still valid this step is skipped.
2. Owner exchanges a session key K_{OA} with its node. If a previous session key is still valid this step is skipped.
 - a) Node and owner know each other's public keys.
 - b) Owner requests a node authentication from the gateway.
 - c) Gateway and Node A mutually authenticate and exchange session key K_{GA} . If a previous session key is still valid this step is skipped.
 - d) Gateway forwards the request to the node.
 - e) Gateway relays authentication sequence between owner and node.
3. The owner requests to transmit data to its node by sending the relevant command and payload. Both the payload and the message are signed by the owner. Message is encrypted using K_{GO} while the payload is encrypted using K_{OA} .
4. Gateway forwards the payload to the node. Communication is encrypted using K_{GA} and signed by the gateway.
5. Response is generated by the node and forwarded to the gateway. Both the payload and the message are signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
6. Gateway forwards the payload to the owner. Communication is encrypted using K_{GO} and signed by the gateway.

Scheme 2:

1. Owner authenticates the gateway and exchanges a session key K_{GA} . If a previous session key is still valid this step is skipped.
2. Owner authenticates Node A using an exchange protected by the master key K_{MOA} or by a derived key. Session key K_{OA} is exchanged or is derived from the master key K_{MOA} . If a previous session key is still valid this step is skipped.

- a) Node and owner know the master key K_{MOA} .
 - b) Owner requests a Node A authentication from the gateway.
 - c) Gateway and owner mutually authenticate and exchange session key K_{GO} . If session key still valid this step is skipped.
 - d) Gateway forwards the request to the owner.
 - e) Gateway relays authentication sequence between owner and node.
3. The owner requests to transmit data to its owner by sending the relevant command and payload. Both the payload and the message is signed by the owner. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
 4. Gateway forwards the payload to Node A. Communication is encrypted using K_{GO} and signed by the gateway.
 5. Response is generated by Node A and forwarded to the gateway. Both the payload and the message is signed by Node A. Message is encrypted using K_{GA} while the payload is encrypted using K_{OA} .
 6. Gateway forwards the payload to the owner. Communication is encrypted using K_{GA} and signed by the gateway.

5.3.3.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.
 - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
 - Non-repudiation is not needed between the gateway on the node.
 - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.4 Node \Leftrightarrow Node: Trusted Gateway

This protocol describes the steps taken when a node needs to communicate with another node in a system where the gateway is trusted. This protocol must be followed each time node-to-node communication takes place so that the gateway can provide proof in case of disputes. Please refer to figure 5.4. Two different protocols are proposed. Scheme 1 requires the node to perform asymmetric cryptography while scheme 2 requires only symmetric cryptography.

5.3.4.1 Methods

Scheme 1:

1. Node A authenticates the gateway and session key K_{GA} is exchanged. If a previous session key is still valid this step is skipped
2. Node A requests to communicate with Node B.
3. Gateway authenticates Node B and session key K_{GB} is exchanged. If a previous session key is still valid this step is skipped.
4. Gateway generates session key K_{AB} .
5. Gateway sends this session key to Node A and Node B encrypting with K_{GA} and K_{GB} respectively. Both messages are signed by the gateway.
6. Node A and Node B authenticate each other and share information using K_{AB} . Messages are signed by the sender.

Scheme 2:

1. Node A authenticates the gateway using an exchange protected by the master key K_{MGA} or by a derived key. Session key K_{GA} is exchanged or is derived from the master key, K_{MGA} . If a previous session key is still valid this step is skipped.

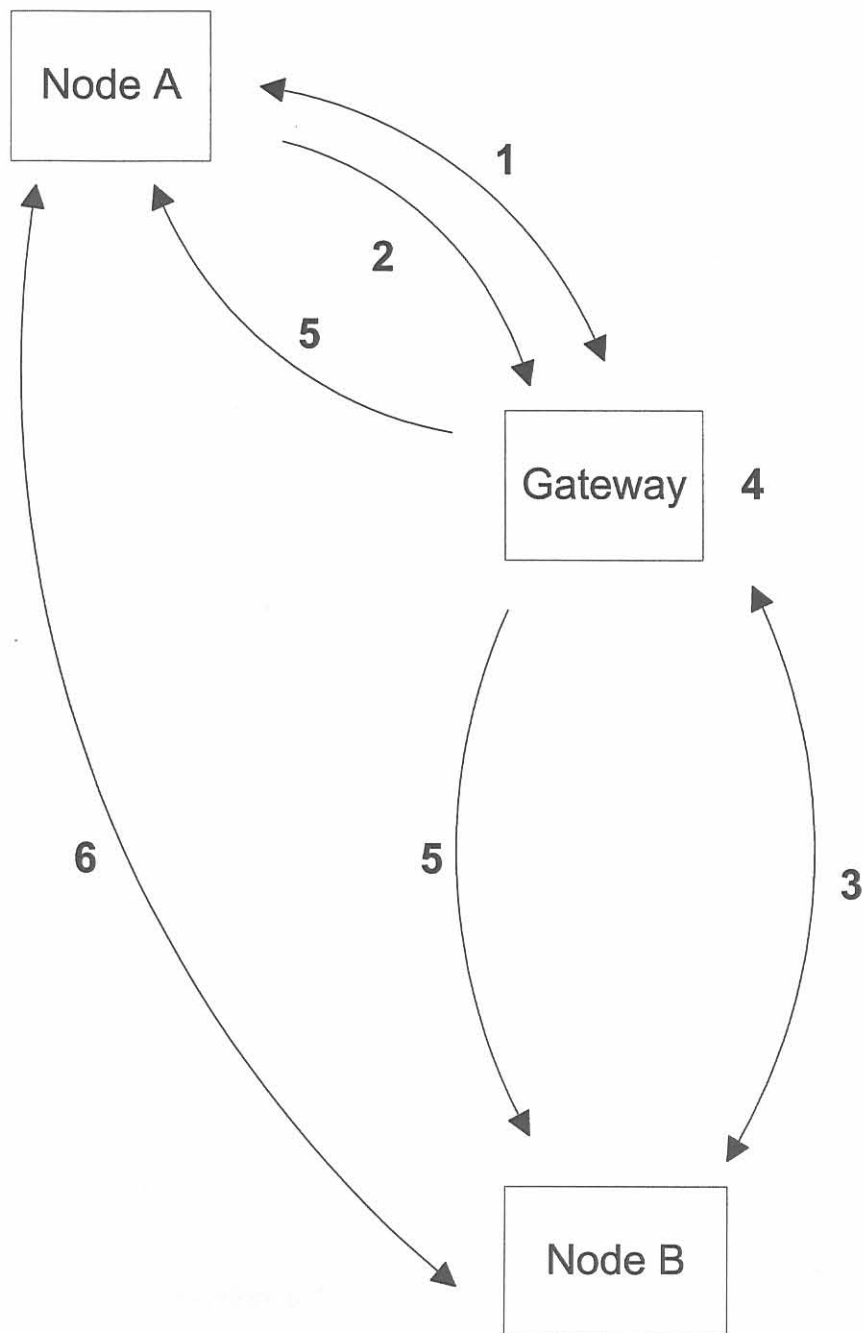


Figure 5.4:

Security protocol for node \Leftrightarrow node communication with a trusted gateway

2. Node A requests to communicate with Node B.
3. Gateway authenticates the gateway using an exchange protected by the master key, K_{MGB} , or by a derived key. Session key K_{GB} is exchanged or is derived from the master key K_{MGB} . If a previous session key is still valid this step is skipped.
4. Gateway generates session key K_{AB} .
5. Gateway sends this session key to Node A and Node B encrypting with K_{GA} and K_{GB} respectively. Node B also receives Node A's access control attributes. Both messages are signed by the gateway.
6. Node A and Node B authenticate each other and share information using K_{AB} . Messages are signed by the sender.

5.3.4.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.
 - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
 - Non-repudiation is not needed between the nodes as the gateway records the transaction.
 - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.5 Node \Leftrightarrow Node: Untrusted Gateway

This protocol describes the steps taken when a node needs to communicate with another node in a system where the gateway is not trusted. This protocol must be followed each time node-to-node communication takes place so that gateway can provide proof in case of disputes. Please refer to figure 5.5.

5.3.5.1 Methods

Scheme 1:

1. Node A send a request to its owner to communicate with Node B. If needed owner is authenticated and session key K_{OA} exchanged. See scheme 1 in section 5.3.3.
2. Owner A authenticates Owner B and session key K_{OAOB} is exchanged. If a previous session key is still valid this step is skipped. Authentication is done using X.511 and CA certificates. See gateway-owner authentication in section 5.3.1.
3. Owner A requests communication possibilities with Node B.
4. Owner B authenticates Node B and exchanges key K_{OB} .
5. Owners decide on session key K_{AB} and Node A's access control attributes are send to Owner B.
6. Owner B sends K_{AB} and Node A's access control attributes to Node B using K_{OB} and signs the message.
7. Owner A sends K_{AB} to Node A using K_{OA} and signs the message.
8. Node A and Node B authenticate each other and share information using K_{AB} . Messages are signed by the sender.

Scheme 2:

1. Node A send a request to its owner to communicate with Node B. If needed owner is authenticated and session key K_{OA} exchanged. See scheme 2 in section 5.3.3.
2. Owner A authenticates Owner B and session key K_{OAOB} is exchanged. If a previous session key is still valid this step is skipped. Authentication is done using X.511 and CA certificates. See gateway-owner authentication in section 5.3.1.

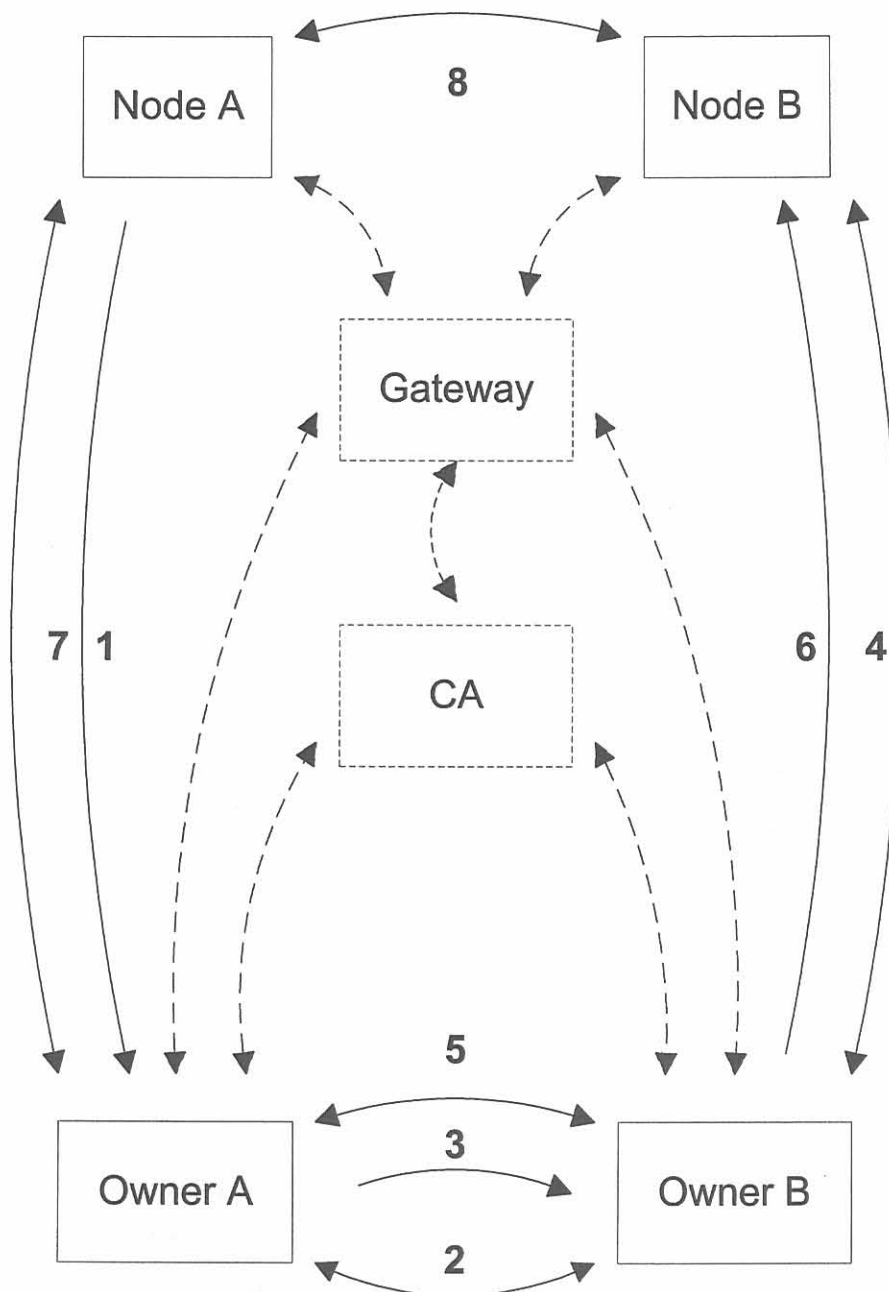


Figure 5.5:
Security protocol for node \leftrightarrow node communication with an untrusted gateway

3. Owner A requests communication possibilities with Node B.
4. Owner B authenticates Node B and exchanges key K_{OB} by using K_{MOB} or a derivation.
5. Owners decide on session key K_{AB} and Node A's access control attributes are send to Owner B.
6. Owner B sends K_{AB} and Node A's access control attributes to Node B using K_{OB} and signs the message.
7. Owner A sends K_{AB} to Node A using K_{OA} and signs the message.
8. Node A and Node B authenticate each other and share information using K_{AB} . Messages are signed by the sender.

5.3.5.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.
 - X.511 authentication procedure recommended.
3. Symmetric key cryptography to provide digital signature.
 - Non-repudiation is not needed between the nodes as the owners record the transactions.
 - A MAC will therefore suffice to sign the message.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.6 Node \Leftrightarrow Node: Direct

This protocol describes the steps taken when a node needs to communicate with another node and the nodes have sufficient means to negotiate the conditions of communication.

5.3.6.1 Methods

Scheme 1:

1. The other node's public key is obtained using either the gateway's (section 5.3.4) or owners (section 5.3.5) as means of distribution.
2. Node A and Node B then authenticate each other and exchange session key K_{AB} .
3. Messages are encrypted using K_{AB} and signed by the sender.
4. Node A and Node B record transactions for non-repudiation and settling of disputes.

Scheme 2:

1. The other node's master key, K_{MAB} , is obtained using either the gateway's (section 5.3.4) or owners (section 5.3.5) as means of distribution.
2. Node A and Node B then authenticate each other and exchange or derive a session key K_{AB} using K_{MAB} .
3. Messages are encrypted using K_{AB} and signed by the sender using K_{MAB} or a derived signing key.
4. Node A and Node B record transactions.

5.3.6.2 Possible Mechanisms

This dissertation is concerned mainly with the cryptographic mechanisms implemented on the private network (i.e. needed to be performed by the node). The performance of public key cryptography on the public network is assumed to be secure and is accepted in the security community. The following mechanisms need to be provided for scheme 1:

1. Symmetric encryption and decryption.
2. Public key cryptography must provide digital signature and key exchange.
 - RSA only available algorithm to provide both.

- X.511 authentication procedure recommended.

The following mechanisms need to be provided for scheme 2:

1. Symmetric encryption and decryption.
2. Digital signature using symmetric cryptography, i.e. MAC.
3. Symmetric authentication sequence.

5.3.7 Key Management

Two different schemes are proposed for each scenario. One allows the nodes to use public key encryption while the other scheme only specifies symmetric mechanisms. Figure 5.6 shows the overall key management for scheme 1 and figure 5.7 shows the overall key management of scheme 2.

5.3.8 Additional Considerations

Some additional features must be taken into consideration after implementation. Each entity is responsible for access control to its features. If an entity requests data the entity supplying the service must ensure that access control rules are enforced. An adequate security audit structure must be put into place to detect security breaches. This could be automated by using commercial solutions but a management structure must also be implemented to ensure that user concerns are addressed. In the event of a security breach there must be a set of predetermined action plans to ensure that the situation is rectified and losses minimized. The actions should at least specify that affected key-pairs be revoked and the overall policy reviewed. The policy must be reviewed regularly and information from the audit and the users taken into consideration.

5.4 MECHANISM IMPLEMENTATION

This section shows possible ways to implement mechanisms mentioned in section 5.3.1 to section 5.3.6. Mechanism implementations are not novel and were obtained from literature [14],[27],[54],[55]. The following mathematical conventions are used:

- $K_{AB}[M]$ Message M is encrypted using the key shared between entities A and B .
- $A(M)$ Message M is signed by entity A using a secret key.

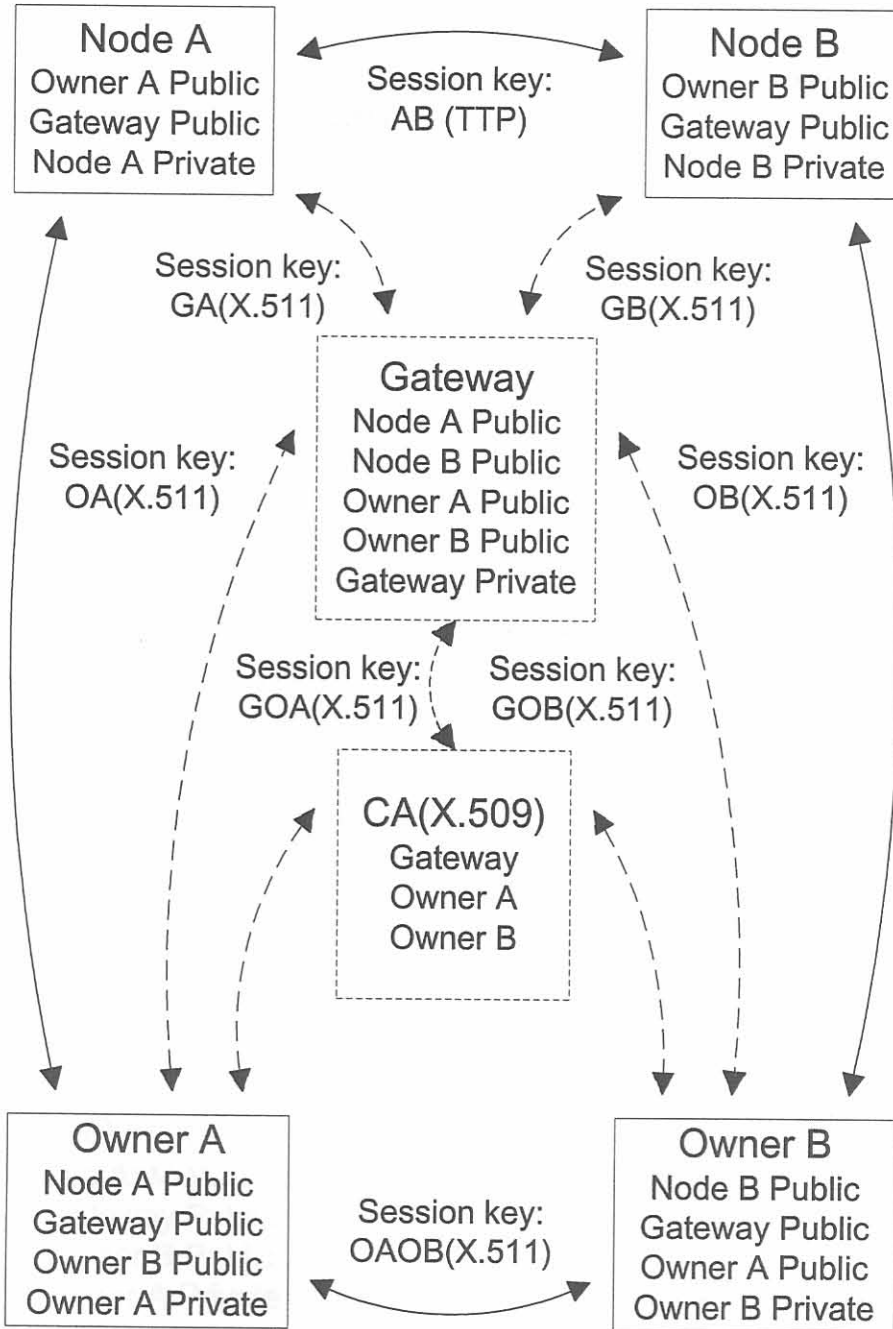


Figure 5.6:
Key management for scheme 1

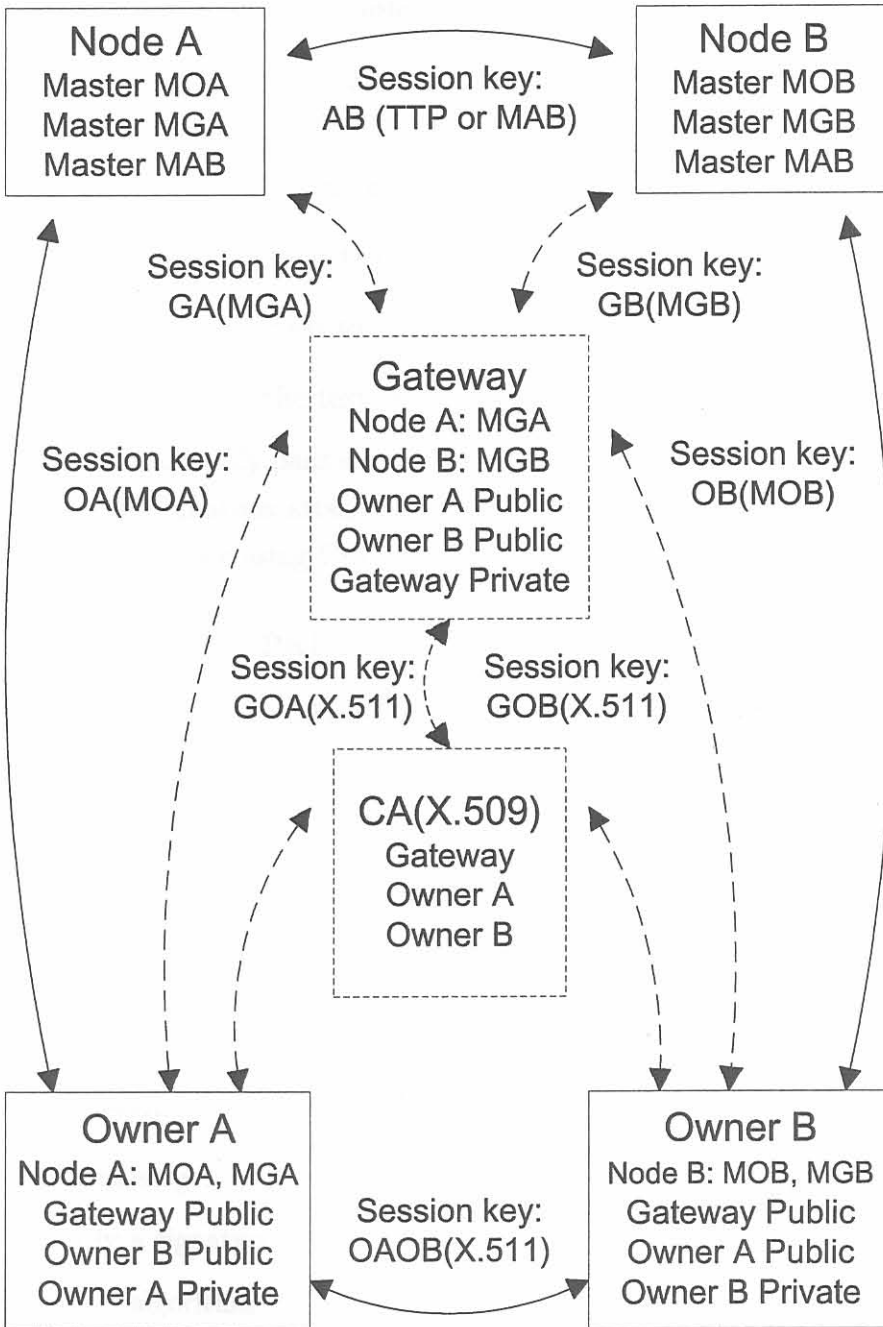


Figure 5.7:
Key management for scheme 2

5.4.1 Confidentiality

Symmetric key cryptography is the easiest mechanism to implement. The steps needed are:

- Select the relevant secret key.
- Select an encryption algorithm, e.g. DES, IDEA, 3DES.
- Select whether encryption or decryption is required.
- Transfer plaintext/ciphertext to smart card.
- Receive the plaintext/ciphertext and store.

The smart card automatically pads a plaintext block to 8 bytes. This padding is random and prevents a crypto-analysis attack that targets small plaintext messages. Successive blocks are chained together using CBC.

5.4.2 Digital Signature: PKI

A digital signature is basically a hash function encrypted with a private key. The steps to compute a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select private key (generally there will be only one).
- Compute the signature (send Compute Signature command).
- Receive the signature and store.
- Append to message.

The steps to verify a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select public key.
- Verify the signature (send Verify Signature command and the signature).
- Check if it successful.

5.4.3 MAC: Symmetric encryption

A digital signature is basically a hash function encrypted with a shared secret key. Smart cards also provide a retail MAC function that used 3DES. The steps to compute a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Select secret key.
- Compute the signature.
- Receive the signature and store.
- Append to message.

The steps to verify a signature are:

- Select a hash algorithm.
- Transfer message data to smart card.
- Receive the hash result.
- Decrypt the received signature using the shared key.
- Compare the two - if they are equal the signature is valid

To be secure the MAC works as follows:

$$MAC(M) = E_{key}[key, h(M)]$$

5.4.4 Authentication: PKI

Authentication using public key cryptography uses digital signatures and must facilitate session key exchange. RSA is the only available algorithm that provides both. The X.511 authentication procedure described in section 2.2.3 is recommended.

One Way Authentication:

$$A \Rightarrow B A(t_A, r_A, ID_B, K_{APublic}[K_{AB}])$$

Two Way Authentication:

$$A \Rightarrow B A(t_A, r_A, ID_B, K_{BPublic}[K_{AB}])$$

$$B \Rightarrow A B(t_B, r_B, ID_A, r_A, K_{APublic}[K_{AB}])$$

Three Way Authentication:

$$A \Rightarrow B A(t_A, r_A, ID_B, K_{BPublic}[K_{AB}])$$

$$B \Rightarrow A B(t_B, r_B, ID_A, r_A, K_{APublic}[K_{AB}])$$

$$A \Rightarrow B A(r_B)$$

The use of timestamps t_X are optional when using three way authentication because sufficient replay attack is provided by the nonces r_X .

5.4.5 Authentication: Symmetric

Authentication using symmetric key cryptography needs not facilitate session key exchange. Session keys are derived from a shared master key distributed by a trusted entity.

The following challenge/response authentication is proposed:

$$A \Rightarrow B A(r_A, ID_A, ID_B)$$

$$B \Rightarrow A B(K_{AB}[ID_A, ID_B, r_A, r_B, DerivedKeyInfo])$$

$$A \Rightarrow B A(K_{AB}[ID_B, r_B, DerivedKeyInfo])$$

5.4.6 Messages

For the system to function correctly a number of message structures need to be defined.

The required message formats are:

- Message format for Node A to gateway:

$$NodeA(K_{GA}[SeqNo, Instruction]NodeA(K_{OA}[Data]))$$

- Message format for gateway to owner:

$$Gateway(K_{GO}[NodeA(K_{OA}[Data]))$$

- Message format for owner to gateway:

$$Owner(K_{GO}[Owner(K_{OA}[Data]))$$

- Message format for gateway to Node A:

$$Gateway(K_{GA}[SeqNo, Instruction]Owner(K_{OA}[Data]))$$

- Message format for Node A to Node B:

$$NodeA(K_{AB}[SeqNo, Instruction, Data])$$

Chapter 6

SYSTEM EVALUATION

6.1 SECURITY ASSURANCE

Proving that a security policy is absolutely secure is impossible. Formal methods, such as BAN logic, can find bugs in a security protocol design as they force the designer to take everything into consideration. These methods are too complicated to prove larger protocols and are also based on educated assumptions. Qualitative and intuitive methods therefore proves to be the most successful when deciding on the effectiveness of a policy. It must also be accepted that no security policy is perfectly secure. Attackers have infinite resources, human errors might create additional vulnerabilities and assumptions made during the implementation might no longer be valid after some time has passed. There are two different concepts that needs to be mentioned [25]:

- Assurance: An estimate of the likelihood that a system will not fail in a practical way. This is based on the process used to develop the system, the experience of the designers and technical assessments. Assurance looks at functionality, strength of mechanisms, implementation and usability. In recent years assurance has placed less emphasis on the product and started to concentrate more on the method used to develop the product.
- Evaluation: The process of assembling evidence that a system meets, or fails to meet, a prescribed assurance target. This is needed when the party relying on the system, and is therefore taking risks using the system, is not the party implementing or designing the system. Evaluation is either performed by the relying party until it is satisfied or by a third party (CLEF). CLEF's audit and test the technical aspects of the system according to a set standard. An evaluation certificate is then issued stating the evaluation assurance level of the policy. There are a number of these standards [25]:
 1. FIPS 140-1: Tamper resistance of cryptographic processors.
 2. IVV: Standards for nuclear weapon systems and NASA manned space flights.
 3. Orange book: Used by the government of the USA
 4. CTPEC: Accepted standard in Canada.
 5. ITSEC: Accepted standard in European countries.
 6. Common Criteria: Single standard now accepted in the US, Canada and Europe. Also see section 2.2.4

To evaluate the system proposed in this report would not be possible taken that the resources are not available to submit it to a CLEF. The best way to determine the effectiveness of the policy would be to use an intuitive and qualitative approach. Assurance will be given that the policy meets a number of criteria defined by the designer. The following criteria must be adhered to:

1. The design methodology followed must adhere to accepted standards or specifications.
2. The implemented policy must assure that the functional requirement of the network model identified in section 3.2.2 will be upheld.
3. The implemented policy must assure that all the threats identified in section 3.2.3 are addressed.

6.2 PERFORMANCE OF SECURITY MECHANISMS

Different security policies were described in section 5.3. Each of these policies specified alternative mechanisms to implement some security services. Some security services can be implemented using different mechanisms, e.g. authentication can be accomplished by asymmetric and symmetric cryptography. The implementation of the mechanisms has an impact on the performance of the system and the effectiveness of the particular policy. The embedded system's main constraint is storage space and not processing power. A policy with an excessive storage requirement cannot be implemented while a powerful workstation and an 8-bit controller can both implement cryptographic mechanisms, the one is just faster. Apart from storage and processing resources a mechanism also requires careful management depending on its complexity and requirements, e.g. symmetric key encryption requires an extensive key distribution system. The following aspects of the security mechanisms must therefore be determined (given in order of importance):

1. Storage requirements: Memory is an expensive commodity and can easily raise the cost of a system a great deal. The variables and storage space required to implement the mechanism must be evaluated.
2. Processing requirements: The time taken to execute the operations needed to implement the mechanism must be evaluated.

3. Other requirements: Additional steps might be needed in the security policy if certain mechanisms are used. Complexity of the mechanism's implementation and management must be taken into account.

6.3 TEST SYSTEM

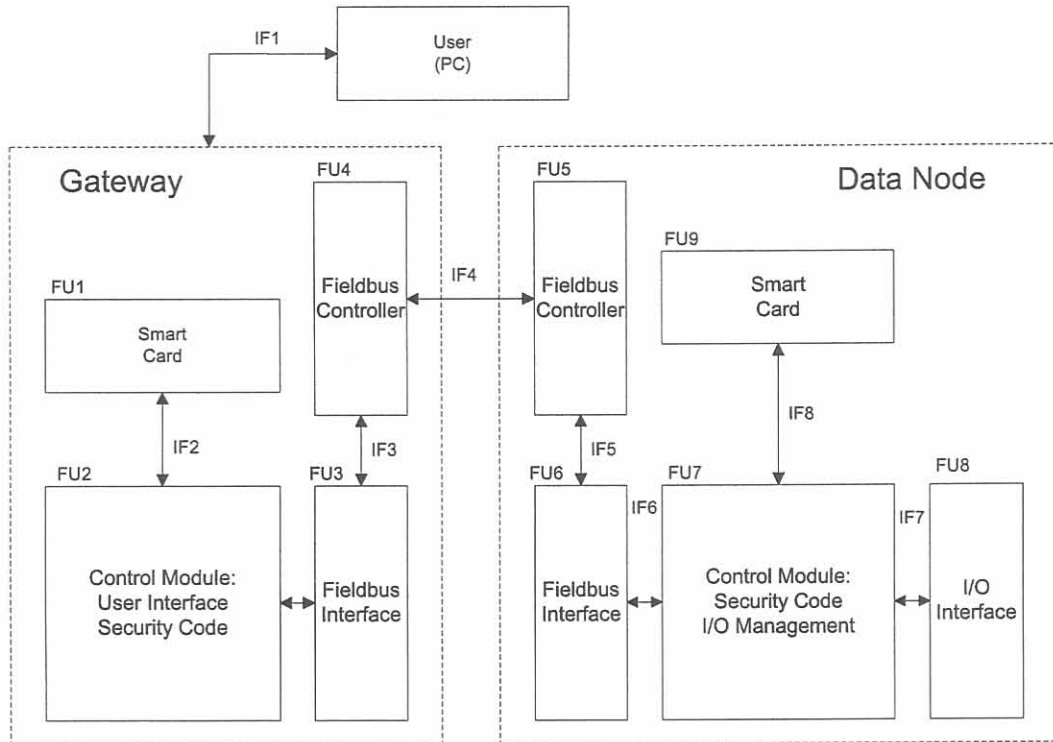


Figure 6.1:
Functional analysis of the test system

A test system was implemented in order to test the performance and viability of the security policies. The functional analysis of the system is shown in figure 6.1. The security module (FU1 and FU9) incorporates a smart card and the support circuitry to connect it to the control modules (FU2 and FU7). The user interface shows statistics and data gained from nodes on the network. It also allows the user to issue commands for various tasks that the node can perform. The PC interface (IF1) physically connects the PC to the rest of the system. The gateway's control module (FU2) regulates information flow between the PC, security module and the network interface (FU3). The network interface and controllers (FU3, FU4, FU5 and FU6) allow the control modules physical access to

the network. The network interface (IF4) interconnects the PC gateway and all the nodes of the system together and allows for information to flow between them. The external interface (FU8) allows the node's control module (FU7) to gain access to external processes. The control module (FU7) interacts with the external interface (FU8) and controls traffic flow to the security module (FU9). It also responds to commands sent over the network interface (IF4) from the rest of the system. The hardware implementation is shown in figure 6.2.

Brief description of system components:

- The CAN protocol was developed in the 1980s for automobile applications. The CAN protocol implements the two lower layers (physical and data link) of the OSI network layer model. CAN is a real-time serial bus protocol offering high reliability, both physically and with error correction [56]. Messages are not addressed to specific stations but a unique identifier labels each message. This identifier describes the message contents and the priority of the message. Only data that is applicable to a station's own function will be processed. This means that only data with an identification header that realises a filter hit on the node will be processed. The CAN protocol is easy to implement as a wide range of IC manufacturers offer CAN bus drivers with DSPs and MCUs being released that have onboard CAN capabilities. The low bandwidth, bus architecture and message based communication models the behaviour of field area networks in general. The system's CAN network is configured at 125Kb per second.
- Microchips's PIC16F876 microcontroller is a 8-bit processor (200ns instruction cycle) with 256 bytes of EEPROM and 768 bytes of data RAM. It contains the peripherals to connect to the smart card, the user PC and CAN interface.
- The smart card used is the STARCOS 2.3 system by Giesecke and Devrient [57]. The smart card communicates at 9600 bps and uses the T=1 communication protocol (see ISO 7816 [37] for details). The cards used had 32Kbytes of data memory. It provides public and symmetric key cryptographic mechanisms:
 - Encryption: DES, 3DES, RSA
 - Signatures: DSA and RSA
 - Hash: 3DES, MD5, SHA-1

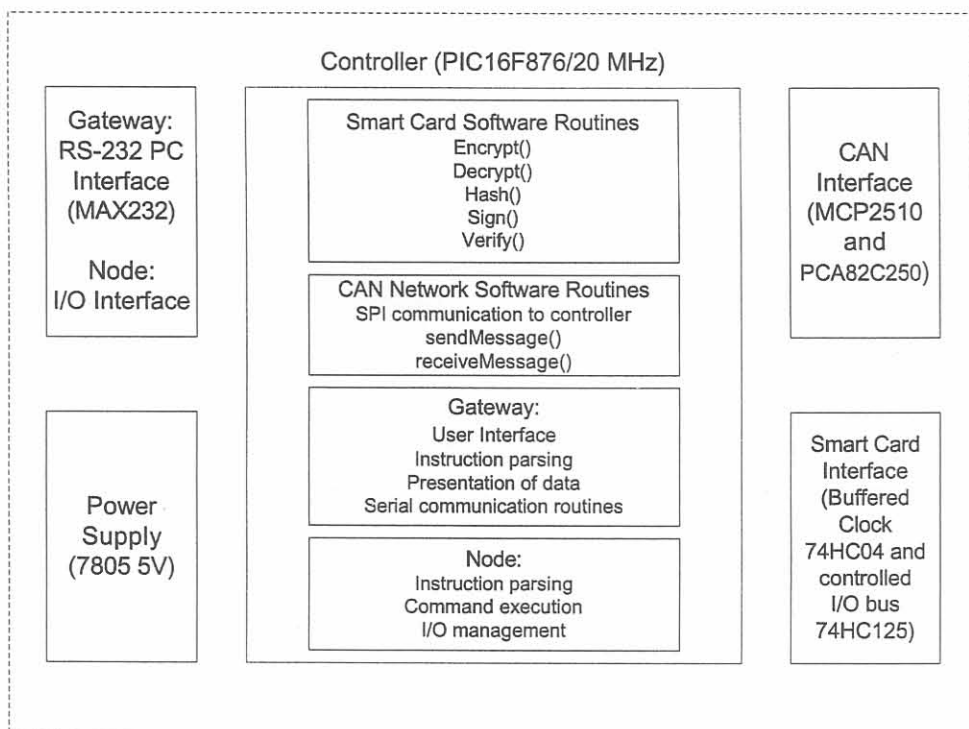


Figure 6.2:
Hardware implementation of the system

Chapter 7

RESULTS

7.1 SECURITY ASSURANCE

Two standards were obtained which provide guidelines for the management of information security: The ISO 13335 [17] and the ISO 17799 [18]. A flow diagram showing the security management as implemented by these standards is shown in figure 7.1.

The methodology followed in these guidelines can be summarized as follows:

1. Establishment of a review boundary:

Prior to asset identification and valuation, the boundaries of the review should be defined. A careful definition of boundaries at this stage avoids unnecessary work and improves the quality of the risk analysis. The boundary description should clearly define the system and any assumptions made. See section 3.1.

2. Risk analysis:

An initial high level risk analysis is performed for the system assets, in each case concentrating on the business values and the serious risks to which the asset is exposed. For all other system components a baseline approach should be chosen to provide general security services. See section 3.2.

- Identification of assets:

As asset is a component or part of a total system to which an organization directly assigns value and hence for which the organization requires protection. See section 3.2.1.

- Valuation of assets and establishment of dependencies:

After listing all assets of the system under review, values should be assigned to these assets. Dependencies of assets on other assets should also be identified, since this might influence the values of the assets and the appropriate level of protection. See section 3.2.2.

- Threat assessment:

Both accidental or deliberate threat sources should be identified. It is essential that no relevant threat is overlooked, since this could result in failure or weaknesses in system security. At the completion of the threat assessment, there will be a list of threats identified, the assets or groups of assets they would affect. See section 3.2.3.

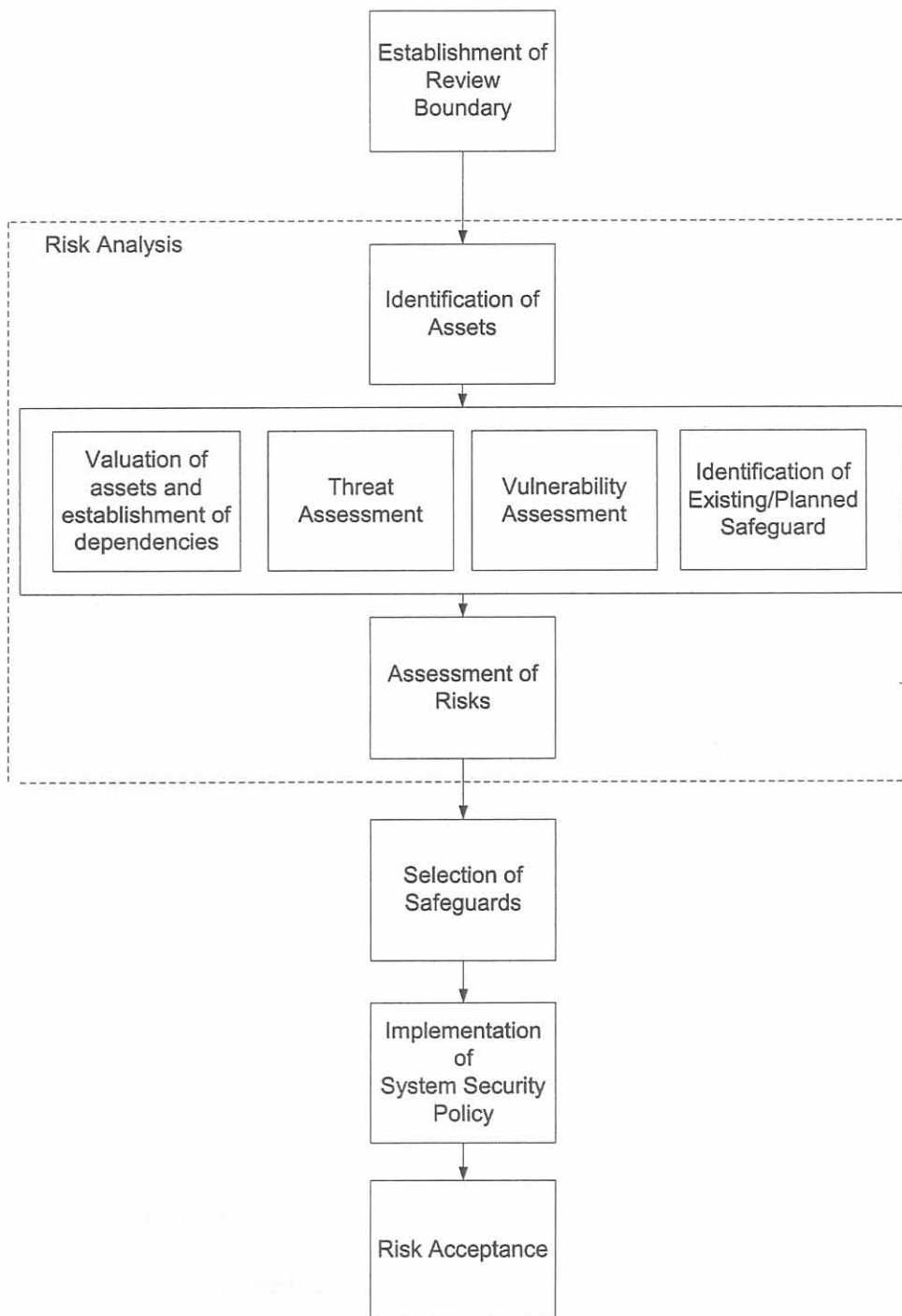


Figure 7.1:
Management of IT security

- Vulnerability assessment:

This assessment includes identifying weaknesses that may be exploited by a threat source to cause harm to the assets or disrupt the system requirements they support. See section 3.2.3.

- Assessment of risks:

The objective of this step is to identify and assess the risk to which the IT system and its assets are exposed. Risks are a function of the values of the assets at risk and the likelihood of threats occurring to cause potential adverse business impacts. See section 3.2.3.

- Identification of existing/planned safeguard:

Possible security services and mechanisms that can safeguard the system are identified. See section 2.2.3.

3. Selection of safeguards:

Appropriate safeguards must be selected to reduce the assessed risks to an acceptable level. The security and system architecture, and constraints of various types must be taken into account to allow for a proper selection. See chapter 4.

4. Implementation of security policy:

Details must be given of selected safeguards and how they must be used to ensure security. A security plan should be suggested that defines the actions that must be taken to implement the required safeguards in the system. See chapter 5.

5. Risk Acceptance:

After evaluating the performance of the security policies residual risks are identified. No system is entirely secure and any implementation will have trade-offs between performance and security assurance. The system managers must review potential adverse effects of these risks and decide whether they are acceptable. This will determine which policy is best to implement. See section 8.2.1.

The design methodology used in this dissertation is shown to cover the same areas as those specified in accepted standards. Therefore it can be concluded that the methodology used satisfies criteria 1 as specified in section 6.1.

In order to show that criteria 2 and 3 are satisfied all the identified threats and functional requirements are mapped to the security services that addresses them. Mechanisms used

to implement these services are well documented and based on proven algorithms. Therefore assurance is given that the implementation is secure and mitigates risk from the defined threats. Table 7.1 shows the implemented security services and the respective threats and functional requirements that they cover.

Table 7.1:
Threat/Functional Requirement vs Service mapping

S1	S2	S3	S4	S5	S6
F8, F9	F5, F6, F7, F8, F9	F4, F11	F10	F12	F1, F2, F3
N6, N7	N2, N4, N5, N6, N7	N3	N2	N8	N1
FAN1	FAN1, FAN2, FAN5	FAN4	FAN2, FAN3		
G2, G3, G6,	G2, G3, G4, G5	PN3, PN4 G6	PN2 G6		PN1 G1
O3, O4	O2, O3, O4	O5	O6	O7	O1

7.2 PERFORMANCE OF SECURITY MECHANISMS

Performance measures were obtained from literature [58], calculations and measurements from the test system. Time taken to perform operations common to all mechanisms, e.g. key selection, has been ignored and only operations unique to the specific mechanisms, e.g. time taken to perform hash, were taken into account. The other operations common to all mechanism, e.g. selection of key, were ignored. These are not important in our analysis, as we are not interested in the detailed performances of the node or the smart card. Measurements and calculation must only provide a means whereby the relative speeds of the different implementation can be compared.

7.2.1 Confidentiality

The DES algorithm was implemented. The results obtained are shown in figure 7.2. DES is a block cipher with block size equal to 8 bytes. This can also be seen from the measurements as the time increase in steps at multiples of 8. Performance measures given by manufacturers are sometimes misleading as some only list the actual processing time. Unfortunately the 9600 bps communication adds additional time as the data must first be transferred and then received back. The smart card might take only $17\mu s$ to encrypt/decrypt a 64 bit block but needs $13.3\mu s$ to transfer the data. In the encryption graph the effect of the data transfer is seen by the skew steps as n increases. Encryption has flat top steps because n is always a multiple of 8.

7.2.2 Digital Signatures

The RSA signatures and the MAC suggested were implemented. The results obtained are shown in figure 7.3. As expected the MAC was faster than the RSA public key signatures. The RSA algorithm however provides better non-repudiation as the MAC can be generated by two parties. It must also be remembered that the MAC is only 196 bits long while the RSA signatures are 512, 768 and 1024 bits respectively. This makes a difference when the signature is sent over a low-bandwidth network with small data packets and high overhead. This also increases the time taken to transfer data to the smart card. Due to the nature of RSA verification takes less time than signing. The MAC has taken the same time for signing and verification.

7.2.3 Authentication: PKI

The three-way authentication procedure was chosen because it does not need time stamps. A synchronized clock is difficult to implement in a distributed system and therefore freshness of messages must be ensured using a nonce. This authentication scheme can easily be performed by the owner and the gateway in a matter of milliseconds. It is therefore only needed to determine whether the nodes can perform this scheme and how long it would take them. Therefore the performance measurements are given for a node to node authentication.

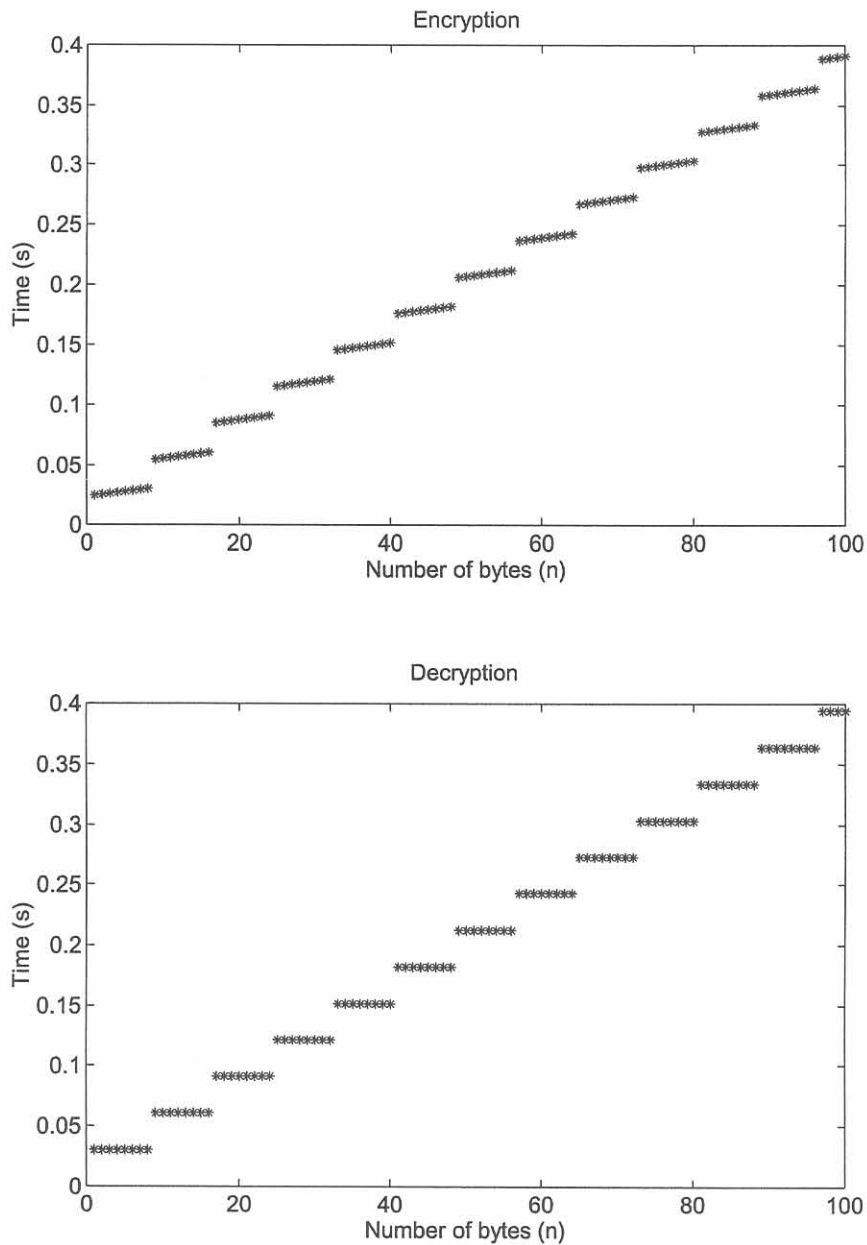


Figure 7.2:
Performance of DES encryption and decryption

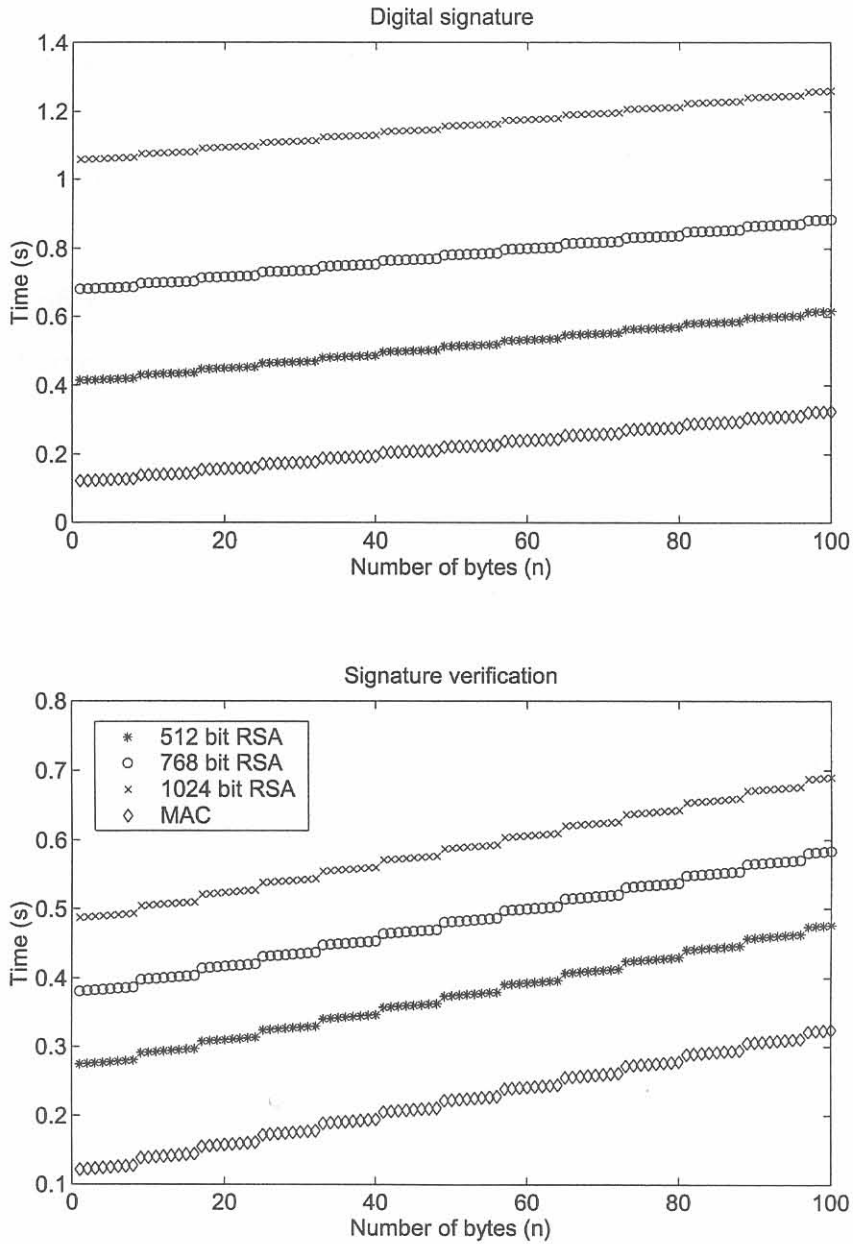


Figure 7.3:
Performance of proposed digital signature mechanisms

Calculations are shown using the RSA - 512 bit algorithm:

1. 1st Step: A to B

- Two random numbers for K_{AB} and r_A : $110ms$
- Encryption K_{AB} : $270ms$
- Signature: $587ms$
- Transmission: $126ms$
- Verification: $393ms$
- Decryption: K_{AB} : $456ms$
- TOTAL: $1.942s$

2. 2nd Step: B to A

- One random numbers for r_A : $55ms$
- Encryption K_{AB} : $270ms$
- Signature: $587ms$
- Transmission: $126ms$
- Verification: $393ms$
- Decryption K_{AB} : $456ms$
- TOTAL: $1.887s$

3. 3rd Step: A to B

- Signature: $456ms$
- Transmission: $54ms$
- TOTAL: $510ms$

4. TOTAL (Steps 1 to 3): $4.339ms$

Given the previous performance measurements of RSA signatures with 768 bit and 1024 bit keys it can be estimated that (with the extra bits needed to be transferred to the smart card and the other node) authentication with the larger keys sizes might take respectively up to 1.5s and 3s longer.

7.2.4 Authentication: Symmetric

This authentication scheme can easily be performed by the owner and the gateway in a matter of μ 's. The owner and the gateway would probably only use PKI when authenticating each other. It is therefore only needed to determine whether the nodes can perform this scheme and how long it would take them. Therefore the performance measurements are given for a node to node authentication.

$A \Rightarrow B A(r_A, ID_A, ID_B)$

$B \Rightarrow A B(K_{AB}[ID_A, ID_B, r_A, r_B, DerivedKeyInfo])$

$A \Rightarrow B A(K_{AB}[ID_B, r_B, DerivedKeyInfo])$

1. 1st Step: A to B

- Random number generation: 55ms
- Signature: 124ms
- Transmission: 23ms
- Verification: 124ms
- TOTAL: 326ms

2. 2nd Step: B to A

- Random number generation: 55ms
- Encryption: 28ms
- Signature: 128ms
- Transmission: 27ms
- Decryption: 30ms
- Verification: 128ms
- TOTAL: 396ms

3. 3rd Step: A to B

- Encryption: 27ms
- Signature: 128ms
- Transmission: 27ms
- Decryption: 30ms

- Verification: 128ms
- TOTAL: 340ms

4. TOTAL (Steps 1 to 3): 1.06s

As expected this system is much faster and the session key does not need to be explicitly updated as many smart cards support build in functionality using derived keys.

7.2.5 Messages

The required message formats are:

1. Message format for Node A to gateway:
 $NodeA(K_{GA}[SeqNo, Instruction]NodeA(K_{OA}[Data]))$
2. Message format for gateway to owner:
 $Gateway(K_{GO}[NodeA(K_{OA}[Data]))$
3. Message format for owner to gateway:
 $Owner(K_{GO}[Owner(K_{OA}[Data]))$
4. Message format for gateway to Node A:
 $Gateway(K_{GA}[SeqNo, Instruction]Owner(K_{OA}[Data]))$
5. Message format for Node A to Node B:
 $NodeA(K_{AB}[SeqNo, Instruction, Data])$

Message 1 requires 2 symmetric encryptions and two signatures. Based on previous results it would be an advantage if symmetric signatures were used. Message 2 requires 1 symmetric encryption and 1 signature. The gateway and owner can implement public key signature easily and effectively and therefore RSA must be used to sign this message. RSA will also give better non-repudiation and integrity protection seen as this message must traverse the public network. Message 3 requires 2 symmetric encryptions and two signatures. Based on previous results it would be an advantage if symmetric signatures were used when signing the payload (taken that the node will have to verify it). The complete message can be signed using RSA and will be verified by the gateway. Message 4 requires 1 symmetric encryption and 1 signature. The node must verify the signature and there a symmetric signature would speed up the communication process. Message 5 requires 1 symmetric encryption and 1 signature. Communication between nodes is not feasible using RSA. The previous messages all had one or more entity that could do public key cryptography efficiently. If both entities are hampered communication is drastically slowed down. Therefore a symmetric integrity and signing mechanism must be used.

Chapter 8

CONCLUSION

8.1 SUMMARY OF THE WORK

The purpose of this dissertation was to show whether smart cards could be used to secure field area network. A literature study provided background on field area networks, the requirements of information security and what smart cards could offer. The system that had to be secured was defined and possible threats were identified. Functional requirements were determined and a risk analysis done to see how the threats could have an impact on the system. Safeguards were selected to provide the security services and two security policy schemes, using different mechanisms, were formulated. Both schemes were evaluated to determine whether security could be assured. Performance was also evaluated to determine whether it is feasible to implement these schemes in a field area network.

8.2 SUMMARY OF THE RESULTS

8.2.1 Effectiveness of Security Policies

8.2.1.1 Scheme 1

Scheme 1 has a major advantage when it comes to key management. Master keys do not need to be kept as session keys can be exchanged at will using RSA. RSA digital signatures also provide better repudiation. This scheme has two major problems:

- Performance: The nodes and the smart cards cannot efficiently implement the PKI functionality. Mutual authentication and key exchanges taking over 4s are not feasible. If the system requires the extra security this scheme can be implemented.
- RSA: Many countries do not allow public key cryptography for the purpose of data encryption (as required by the X.511 protocol) but only as a means to perform digital signatures. Therefore many smart cards cannot recover data encrypted using RSA. The digital signature basically uses RSA decryption for signing and encryption for verification. Verify signature commands implemented in smart cards do not return any data.

8.2.1.2 Scheme 2

Scheme 2 has a slightly more intricate key management system, as nodes, gateways and owners need to keep updated symmetric keys. The key management suggested never allows the master key shared between the owner and the node to be revealed (it is never send over a data link). Therefore owner to node communication should be very secure.

The other master keys are also distributed using secure channels. If a node's master key with its owner should be compromised it can be revoked and only a new smart card is needed to restore functionality. This system also places fewer burdens on the node's processing and storage resources. This scheme has one major problem:

- Non-repudiation: The symmetric nature of the signature does not allow for non-repudiation as both the receiver and the sender can generate the signature. The transaction must therefore be recorded by an independent arbiter. This is however only an issue with node-to-node communication. The way the key management is done requires the node to contact the gateway or its owner to gain a key. These entities can record the nature of the transaction and settle later disputes.

8.2.2 Comparison of Security Policies

Both the proposed schemes adequately provide the necessary security functions. Although scheme 2 does not have such a high security assurance as scheme one it definitely performs better using low-resource hardware. It is suggested that scheme 1 is utilized in advanced field area networks where nodes have sufficient resources to implement PKI efficiently e.g. factory and industrial automation. Scheme 2 should be used to secure simple field area networks in less intensive environments, e.g. remote data loggers or power measurement.

8.3 CONCLUSIONS

Finally, the following important conclusions can be made:

- It is possible to formulate a policy which could be implemented successfully into a field area network using the security mechanisms provided by a smart card while maintaining reasonable system performance, despite the resource limitations of the field area network.
- The performance of the security operations, as implemented using smart cards, is adequate but application specific considerations must still be taken into account regarding the scheme to be applied. Both schemes assures security, but scheme 1 has the major advantage of easier key management at the cost of performance, while scheme 2 performs better at the cost of a more intricate key management system. Which scheme to use is up to the implementer and application constraints.

- The performance measures of field area networks are fairly predictable. This enables the designer to determine the time to perform security operations in advance and then make a suitable policy choice accordingly.

8.4 SUGGESTIONS FOR FUTURE WORK

Future work should include the following:

- Evaluation and verification of the proposed security policies using cryptanalysis and formal methods.
- Algorithm design for smart cards. Faster digital signatures and key exchange using public key cryptography.

Chapter 9

REFERENCES

- [1] D. Dietrich and T. Sauter, "Evolution potentials for fieldbus systems", *Proceedings IEEE International Workshop on Factory Communication Systems*, invited paper, Porto, Portugal, Sep. 2000, pp. 343-350.
- [2] S. Suzuki and K. Nakada, "An authentication technique based on distributed security management for the global mobility network", *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 8, Oct. 1997, pp. 1608-1617.
- [3] C.A.C.M. Couto, J.C. Metralho and C.M.J.A. Sercada, "CAN based actuator system for greenhouse control". *Proceedings of the ISIE'99 IEEE International Symposium on Industrial Electronics*, vol. 2, July 1999, pp. 945-950.
- [4] J. Al-Muhtadi, M. Anand, M.D. Mickunas and R. Campbell, "Secure smart homes using Jini and UIUC SESAME", *16th Annual Conference on Computer Security Applications*, 2000, pp. 77-85.
- [5] B. Courbun, "Paying for energy the smart way", *IEE Review*, vol. 47, no. 4, July 2001, pp. 17-20.
- [6] V.M. Cordonnier, "Smart cards: present and future applications and techniques", *Electronics and Communication Engineering Journal*, vol. 3, no. 5, Oct. 1991, pp. 207-212.
- [7] J.F. Dhem, D. Veithen and J.J. Quisquater, "SCALPS: Smart card for limited payment systems", *IEEE Micro*, vol. 16, no. 3, Jun. 1996, pp. 42-51.
- [8] A. Kara, "Secure remote access from office to home", *IEEE Communications magazine*, 2001, pp. 68-72.
- [9] M. Chang, "Security on database systems and distributed databases", *25th Annual 1991 IEEE International Carnahan Conference on Security Technology*, Oct. 1991, pp. 219-220.
- [10] P. Samarati and R. Sandhu, "Authentication, access control and audit", *ACM Computing Surveys*, vol. 33, no. 4, March 1996, pp. 241-243.
- [11] Y. Lee, "Integrating access control with user authentication using smart cards", *IEEE Transactions on Consumer Electronics*, vol. 46, no. 4, Nov. 2000, pp 943-94.
- [12] J. Borst, B. Preneel and V. Rijmen, "Cryptography on Smart Cards", *Elseviers Computer Networks*, vol. 36, 2001, pp. 423-435.

- [13] P.H. Hartel and L. Moreau, "Formalizing the safety of Java, the Java virtual machine and Java card", *ACM Computing Surveys*, vol. 33, no. 4, Dec. 2001, pp. 110-143.
- [14] W. Stallings, *Cryptography and Network Security*, 3rd Edition, Prentice-Hall, 2003.
- [15] ITU-T X.800, *OSI Security, Structure and Applications*, 1991.
- [16] ISO 7498, *Information technology - OSI Basic Reference Model*, 1994.
- [17] ISO 13335, *Guidelines for the management of IT Security*, 1998.
- [18] ISO 17799/BS7799, *Information technology - Code of practice for information security management*, 2000.
- [19] H.K. Chang, K. Farn, C.H. Lu, "Security mechanisms for distributed systems", *IEEE 28th Annual 1994 International Carnahan Conference on Security Technology*, Oct. 1994, pp. 72-75.
- [20] J.K. Lee, S.R. Ryu and K.Y. Yoo, "Fingerprint-based remote user authentication scheme using smart cards", *Electronics Letters*, vol. 38, no. 12, June 2002, pp. 554-555.
- [21] M.H. Sherif, *Protocols for Secure Electronic Commerce*, CRC Press, 2000.
- [22] R. Oppliger, *Security Technologies for the World Wide Web*, 2nd Edition, Artech House, 2003.
- [23] C. Fritzner, L. Nilsen, A. Skomedal, "Protecting security information in distributed systems", *1991 IEEE Computer Society Symposium on Research in Security and Privacy*, May 1991, pp. 245-254.
- [24] V. Hassler, *Security Fundamentals for E-Commerce*, Artech House, 2001.
- [25] R. Anderson, *Security Engineering*, Wiley, 2001.
- [26] ITSEC, www.itsec.gov.uk
- [27] H.P. Konigs, "Cryptographic identification methods for smart cards in the process of standardization", *IEEE Communications Magazine*, vol. 29, no. 6, Jun 1991, pp. 42-48.
- [28] Thawte, www.thawte.com

- [29] K.B. Lee and R.D. Schneeman, "Internet based distributed measurement and control application". *IEEE Transactions on Instrumentation and Measurement*, vol. 48, no. 2, June 1999, pp. 23-27.
- [30] K.B. Lee and R.D. Schneeman, "Distributed measurement and control based on the IEEE 1451 smart transducer interface standards", *IEEE Transactions on Instrumentation and Measurement*, vol. 49, no. 3, June 2000, pp. 621-627.
- [31] H.L. Beon, W.K. Dae, S.K. Hang, J.C. Yang and M.J. Jang, "Analysis of network based control system using CAN protocol", in *Proceedings ICRA IEEE Conference on Robotics and Automation*, vol. 4, 2001, pp. 3577-3581.
- [32] P.W. Halliden, "Security for distributed application", *European Convention on Security and Detection*, May 1996, pp. 156-160, pp 156-160.
- [33] P. Bergstrom, K. Driscoll and J. Kimball, "Making home automation communications secure", *Computer*, vol. 34, no. 10, Oct. 2001, pp. 50-56.
- [34] H.Nakakita, K. Yamaguchi, M. Hashimoto, T. Saito, M. Sakurai, "A study on secure wireless networks consisting of home appliances", *International Conference on Consumer Electronics, ICCE, 2002 Digest of Technical Papers*, 2002, pp. 379-381.
- [35] W.A. Arbaugh and L. van Doorn, "Embedded security: challenges and concerns", *Computers*, vol. 34, no. 10, pp. 40-41, Oct. 2001.
- [36] J.F. Dhem and N. Feyt, "Hardware and software symbiosis helps smart card evolution", *IEEE Micro*, vol. 21, no. 6, Nov. 2001, pp. 14-25.
- [37] ISO 7816, *Information Technology - Identification cards - Integrated circuit(s) cards with contacts*, 1999.
- [38] R. Dettmer, "Getting smarter", *IEE Review*, vol. 44, no. 3, May 1998, pp. 123-126.
- [39] P. Peyret, G. Lisimaque and T.Y. Chua, "Smart cards provide very high security and flexibility in subscribers management", *IEEE Transactions on Consumer Electronics*, vol. 36, no. 3, Aug. 1990, pp 744-752.
- [40] T.S. Messerges, E.A. Dabbish and R.H. Sloan, "Examining smart-card security under the threat of power analysis attacks", *IEEE Transactions on Computers*, vol. 51, no. 5, May 2002, pp. 541-552.

- [41] S.B. Guthery, "Java card: Internet computing on a smart card". *IEEE Internet Computing*, vol. 1, no. 1, Feb. 1997, pp. 57-59.
- [42] J. Elliott, "The MAOS trap". *Computing and Control Engineering Journal*, vol. 12, no. 1, Feb. 2001, pp. 4-10.
- [43] Sun Microsystems: Javacard, <http://java.sun.com/products/javacard/>
- [44] Java Card Forum, www.javacardforum.org
- [45] VISA Forum, www.visa.com
- [46] Global Platform Consortium, www.globalplatform.org
- [47] Microsoft, www.microsoft.com
- [48] MULTOS, www.multos.com
- [49] D. Chadwick, "Smart cards aren't always the smart choice". *Computer*, vol. 32, no. 12, Dec. 1999, pp. 142-143.
- [50] D.B. Parker, "Using threats to demonstrate the elements of information security", *European Convention on Security and Detection*, May 1995, pp. 11-17.
- [51] S.V. Wunnava, E. Lule, "Distributed security schemes for networks", *IEEE South-eastCon 2001*, 2001, pp. 114-117.
- [52] B. Thuraisingham and W. Ford, "Security constraint processing in a multilevel secure distributed database management system", *IEEE Transactions on Knowledge and Data Engineering*, vol. 7, no. 2, Apr. 1995, pp. 274-293.
- [53] J. Verschuren, R. Govaerts and J. Vandewalle, "Simultaneous enforcement of the Bell-LaPadula and the Biba security policy models in an OSI-distributed system", *Communications on the Move, Singapore ICCS/ISITA '92*, vol. 1, Nov. 1992, pp. 257-263.
- [54] D. Naccache and D. M'Raihi, "Cryptographic smart cards", *IEEE Micro*, vol.16, no. 3, Jun. 1996, pp. 14, 16-24.
- [55] D. Nyang and J. Song, "Symmetric identity-proving protocol for smart cards", *Electronics Letters*, vol. 35, no. 23, Nov. 1999, pp. 2022-2024.

- [56] M. Barbossa, M. Farsi and K. Ratcliff, "An overview of controller area network".
Control Engineering, vol. 10, no. 3 June 1999, pp. 113-120.
- [57] Starcos 2.3 Manual, Giesecke and Devrient, 2003.
- [58] W. Rankl and W. Effing, *Smart Card Handbook*, Wiley, 2001.