High Security Java Based Architecture For P-NET Internet Applications

Mikhail Gordeev
Institute of Computer Technology
Vienna University for Technology, Austria
mischa@ict.tuwien.ac.at

Abstract - This is already a reality that security aspects play a key role mostly in any information technology application of today. Fieldbus – Internet applications and therefore P-NET Internet applications are not an exception. That is a reason why it is necessary to take security considerations in an account whilst developing any P-NET application dealing with a communication over the Internet. First of all, this paper will make a major accent on a definition of main data security concepts, which must be implemented in such applications. Then we shall point out main groups of threats to their security and concentrate on main requirements to the architecture which should let to avoid them.

I. DATA SECURITY CONCEPTS AND DEFINITIONS

First of all, it is necessary to define what is the meaning of the term "security" for an application, which has to be developed. In the area of information technologies, the term "security" is used in the sense of minimising the vulnerabilities of the information system assets and resources. In other words, it means that all system weaknesses, which could be exploited to violate a system or the information it contains, must be minimised or even eliminated. Their specification is an objective of the special technique called *treat*[1] *identification and risk analysis*, which is an essential procedure for any secure system development process.

Any information recourse of a system can be given a number of properties, which could be a target for threats. We can define the following data main properties, which violation could make a strong negative impact to a system:

- *Availability*. It means that data must be always available when they are needed.
- *Integrity*. It means that data must not be altered or destroyed in an unauthorised manner.
- *Confidentiality*. It means that information is not made available or disclosed to unauthorised individuals, entities or processes.

We also have to mention another rather important data property called non-repudiation, which means that data origin and delivery are proofed, their sending and receiving cannot be falsely denied.

For any system, a *security policy* must be specified before the system specification and development is started. The security policy contains an informal description of the security principles and practise to be used and usually is written in a natural

[1] In this context the term "threat" means a potential violation of a system security

language. A security policy must be written for any system, otherwise it will be completely impossible to understand what security is needed in the system, and therefore it will be not possible to implement it during further development stages. Essentially, a security policy states, in general terms, what is and is not permitted in the field of security during the general operation of the system. A policy sets the topmost level of a security specification.

Now, after we have defined what a security policy is, it is a right time to give a brief description of main concepts and techniques used to satisfy it. The second part of the OSI Reference Model [ISO89] extends the Model by a definition of security architecture for open system interconnection. It specifies a number of *security services*, which could be provided optionally within the OSI Reference Model framework. However their implementation is necessary for any system, which is intended to be secure. We shall give their brief description within the following paragraphs.

- *Authentication*. Services of this group carry out authentication of a communicating peer entity and the source of data.
- *Access control.* This service provides protection against unauthorised access to system recourses. This service can be applied to various types of access like the use of communication recourse, reading, writing or deleting. It may be also applied to all accesses to a resource.
- *Data confidentiality*. Services of this group protect data from unauthorised disclosure. Some of the functions they provide are connection confidentiality, selective field confidentiality and traffic flow confidentiality.
- *Data integrity*. Services of this group protect an information flow from such threats like data modification, insertion, deletion or replaying.
- *Non-repudiation*. The main purpose of these services is to provide a recipient (or a sender) of data with proof of data origin (or delivery). It is done in order not to let a communication entity to falsely deny a fact of data sending or receiving.

The standard also defines particular *security mechanisms*, which can be used to implement certain security services or combinations of them. They can be grouped in the following way:

- *Encipherment.* Encipherment is used to provide data or information flow confidentiality. It can be based on a rather big number of encipherment algorithms [Sch96], which can be divided to two main groups: symmetric (secret key) algorithms and asymmetric (public key) algorithms. Encipherment can be also used in some other security mechanisms.
- *Digital signature mechanisms.* These mechanisms incorporate two main procedures: signing a data unit and verifying a signed data unit. The essential characteristic of a digital signature mechanism is that only using the signer's private key can produce the signature. We can recommend an interested reader to refer to [Sch96] where he can find a complete and detail description of digital signature techniques.

- *Access control mechanisms.* These mechanisms utilise authentication information on a communication entity to determine and enforce access rights of the entity. Usually access control mechanisms are based on use of access control information bases containing data on the access rights of peer entities, authentication information like passwords, security labels and so on.
- *Data integrity mechanisms.* These mechanisms cover two main functions: the integrity of a single data unit and the data stream integrity. Their main task is to protect the transmitted information against modification, manipulation and replay.
- *Authentication exchange mechanism.* The main task of this mechanism is to carry out entity authorisation functions.
- *Traffic padding mechanism.* This mechanism can be used to provide various levels of protection against traffic analysis. However this mechanism can be effective, if it is used in a combination with the confidentiality service.
- *Routing control mechanism.* The main task of this mechanism is to choose secure routes for data communication. For example, if a treat was detected on a certain route then a data flow should be redirected to another route. The security policy also could forbid a communication through certain networks or links. To provide a relevant routing, is also a task of this mechanism.
- Notarisation mechanism. This mechanism introduces in a system a so-called trusted third party. The main objective of which is to assure properties of the data communication between two or more entities.

II. POTENTIAL THREATS TO P-NET INTERNET APPLICATIONS SECURITY

Let's have a look now at a general architecture of the P-NET – Internet application. We can intuitively distinguish its two main parts. The first one is an Internet gateway carrying out a number of functions (they will be described later). The second one is a remote instance (e.g. a management and visualisation application running at a personal computer) performing an access to the P-NET network over the Internet.

In real control applications, we often have to deal with a security sensitive information. A good example could be a remote plant control. In this case, we must deal with a control data flow containing data from the plant and also some control command sent by a remote management system. Do doubts, data security aspects play in this case a dominant role.

In the following paragraphs, we shall point out and examine groups of possible treats to data security of a typical P-NET Internet application. The following picture demonstrates a general representation of such a systems and shows possible targets for threats.

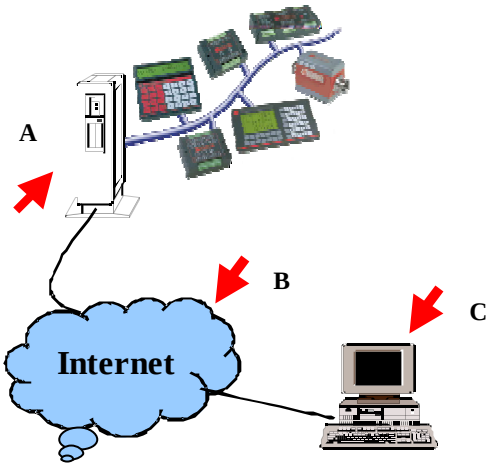

Figure 1: Potential datathreats to a P-NET Internet application

We could point out five main groups of threats:

- *Group A.* Threats of this group are targeted at the Internet gateway from outside. An authorised access to its services could give an intruder a big number of possibilities to break into the system or even take control over the whole plant.
- *Group B.* We think it is not necessary to say that the Internet is a completely insecure area. It is a well-known fact. It is a main reason why data transmitted over has to be fully protected threats to data integrity and confidentiality.
- *Group C.* A personal computer used to access the P-NET network over the Internet can be attacked as well. The threat sources can be different. It could be for example a Trojan horse application (for example a Java applet downloaded from another host). Threats of this group contain a very dangerous potential. We can show it by an example. In order to provide data communication integrity and confidentiality, certain encryption mechanisms can be exploited. File system of the computer could be chosen as good place to store relevant encryption keys. But what will happen, if an intruder "steals" them by means of certain treats? It will mean that the communication entity (in this case the personal computer) will be compromised and an intruder will gain a full[2] access to the P-NET network through it. This treat could be found out only by the proper security management means, of course, if they are provided in the system.

III. SECURE ARCHITECTURE ELEMENTS

A high level of data security can be achieved only by combining a variety of different security mechanisms and techniques. A major accent in any secure control network architecture development must be done on following the guidelines and

[2] As full as it is defined by the entity security policy

implementation of the Security Architecture for Open System Interconnection [ISO89], which was already discussed in this paper. Such an implementation can be reached, as we already said, by utilising different techniques and technologies. In this section of the paper, we shall focus on such of them as secure Internet management protocols, smart card and Java technology. They are supposed to be building blocks of the security architecture proposed in this paper.

Secure Internet management protocols

Specifying an Internet application, we always must make a choice - either we take a standard protocol as a basis or we implement our own one. The first approach has a big number of advantages despite a probable complexity of its implementation. Standard protocols are supported by a great number of software and hardware manufactures, they are proofed by time and thousands of already existing implementations. Their have another rather important advantage. It is inherited in a fact that a system using them can be very easily integrated in already existing applications.

Simple Network Management Protocol (SNMP) is regarded as one of the most promising protocols of this group. Since its first publication in 1988, it has become the most widely used network management tool for TCP/IP based networks. The Simple Network Management Protocol:

- Defines a protocol for exchanging information between one or more management systems and a number of agents
- Provides a framework for formatting and storing management information
- Defines a number of general-purpose management information variables or objects.

Usually, the model of network management used in SNMP deals with such key elements as management station, management agent, management information base (MIB) and network management protocol. This model perfectly fits to requirements to fieldbus Internet applications. We shall leave detail description of integration of SNMP to them outside the scope of this paper and concentrate on security facilities of the latest version of SNMP (SNMPv3).

Security functions of any SNMPv3 engine are implemented in two subsystems called Security Subsystem and Access Control Subsystem. A major task of the Security Subsystem is to perform authentication and encipherment mechanisms. Access Control Subsystem provides authorisation services to control access to management information base (MIB) for the reading and setting of the management objects. The following paragraphs will give a more detail description of these subsystems.

An implementation of a Security Subsystem may support one or more security models. Current version of SNMPv3 defines a User-based Security Model (USM) [RFC2274]. In general, USM protects against the following potential threats to data confidentiality and integrity [Sta98]:

- *Modification of information.* An unauthorized party could alter messages sent from one SNMP entity to another one. This can result on unauthorized management operations like reading and setting of management objects values including those, which are related to configuration and accounting.
- *Masquerade.* A third unauthorized party could act on behalf of an authorized SNMP entity in order to perform certain management operations.
- *Message stream modification*. Due to the fact that SNMP operates over connectionless protocol, another possible treat could be to delay messages, reorder or to replay them. This could result in a rather strong negative impact on the overall managed system functionality. For example, a reset device command can be "recorded" and then "played" once again later.
- *Disclosure.* A third unauthorized party can listen an information flow between SNMP entities and therefore gain some knowledge on managed objects and their values. This type of threat can be quite critical in some cases. For example, lets imagine that one of the managed objects contains an information on home security system. Then an intruder can easily use this information to break into the home.

Concluding a description of the SNMPv3 security model, we shall say few words about cryptographic functions used to implement it.

USM is based on two cryptographic functions: authentication and encryption. These functions can be used separately or in conjunction with each other. Encryption function is based on a symmetric algorithm DES. Usage of this algorithm indents that the same secret key is used, and therefore shared, by SNMP entities involved to the communication process. This secret key serves as an input to the encryption algorithm and is used for both message encryption and decryption. It means that a message can be decrypted only with a key, which was used to encrypt it. Two authentication protocols (HMAC-MD5-96 and HMAC-SHA-96) are used to produce a message authentication code. In other words, such a code could be also called a message digest – digital "fingerprint" of a message. In order to produce such a digest, HMAC takes a message and also a key (called authentication key) as an input and then passes it through a secure hash function. Result of this operation is a code of the fixed size[3], which fully identifies the message. Then the digest is attached to the message and sent to a recipient. When this message is received, it is passed through the authentication algorithm using the same secret authentication key once again. Then the original and just computed digests are compared. If they are the same it means that the message was not altered on its way and it was sent by the alleged sender.

On one hand, usage of two different keys for encryption and authentication purposes makes the system more complex because a need in proper key distribution and management occurs. On another hand, this fact makes the security model more flexible and provides more functionality.

[3] Usually it is 96 bits

Access control is another security function provided by the SNMPv3 framework and implemented by the Access Control Subsystem. This function is performed at the PDU level. In general, an access control function defines mechanisms for determining whether access to a managed object in a local MIB by a remote entity should be allowed or not.

Concluding this section, we would like to say that SNMPv3 framework provides already flexible and sufficient security mechanisms, which could be used to assure a rather high level of data security for systems implementing it. From this point of view, its usage in P-NET Internet applications seems to be more that just promising.

Smart card technology

A smart card could be defined as a "portable data storage device with intelligence and provisions for identity and security" [RaEf99]. An initial application area of the smart cards was banking systems. However, since some years they are used almost in any application having high requirements to its data security and are well known as a secure technology. Due to the closed architecture and implemented cryptographic mechanisms, the smart card is mostly protected against external security attacks. Modern smart cards are also protected against internal attacks as their operational environment could be shared by a number of different applications. Security sensitive data of each application in this case are fully protected from any unauthorised access which could be done by another application running on the same card.

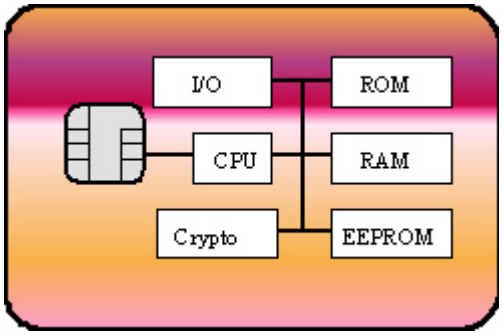

Figure 2: General architecture of a smart card

Physically, a smart card is a plastic card having one or more semiconductor devices embedded in it (fig. 2). The semiconductor device embedded in a true smart card is a microcontroller. Most of the modern smart cards are equipped additionally with a special dedicated cryptographic co-processor. The microcontroller performs computational operations and decision making. It also handles any access to protected storage areas located on the card. The smart card microcontroller consists at a minimum of a central processing unit (CPU), Random Access Memory (RAM), Read Only Memory (ROM) and non-volatile memory.

The main benefit of using smart card technology is increased data security. Micrcontrollers used in smart cards are especially designed to accomplish it. High data security in smart cards is provided by different levels of security varying from

simple PIN numbers to full biometrics, or from simple algorithms to Data Encryption Standard (DES) or to Rivest, Shamir, Adelman (RSA) encryption. For example, in financial applications, a device can use PIN numbers or fingerprints or some other biometrics to verify users. The card is able to verify the reader and the reader is able to verify the card due to the microcontroller ability to perform certain algorithms.

However we have to mention that a smart card, like any other tool used to increase data security, cannot guarantee a complete protection against security attacks [Lemm98]. This fact can not be considered as a serious barrier on the way of smart card application development because we always have to compare the estimated efforts and costs aimed to increase an application security with efforts, value of the protected data and costs of breaking into the system. Taking this fact into consideration, smart card appliances are used to be the most efficient and convenient way to reach a high level of applications security.

IV. BUILDING A SECURE ARCHITECTURE

In this paragraph, we shall provide a general overview of the secure architecture for P-NET Internet applications and formulate main guidelines to their implementation. One of the key aspects of the implementation is used software tools. It is mostly impossible to figure out an "universal" programming language, which could be suitable for any application domain or even a certain application. Usually the choice must be done according the objective requirements to the system, which must be developed. On another hand, functionality, constrains and support[4] of the language must be carefully analyzed and taken into an account.

From all these points of view, Java technology seems to be very promising. It provides a number of rather significant advantages like complete platform independence[5], and therefore portability, application code safety and integrity, support by thousands of software and hardware manufacturers all over the world. Nowadays, Java technology is rapidly developing towards embedded systems and architectures, which is certainly interesting for the world of control networks. Another major advantage of Java is a solid security concepts, which was significantly improved in the latest version of Java known as Java 2.0. However we must also keep in mind some disadvantages of Java, which could be very critical for certain types of applications, mostly control applications. First of all, Java applications require a Java Virtual Machine (JVM), on top of which there are run. It means that requirements to the system resources are rather high. We could also mention some other disadvantages like quite low performance and low reliability.

However despite all these facts and taking the major advantages into an account, we think that Java could be the most suitable technology for implementation of fieldbus – Internet applications. We shall talk about it the next sections in more detail.

---

[4] development tools and software libraries. However in some cases it could be also a special dedicated hardware

[5] with certain limitations on the level of hardware access

Secure architecture in general

Now it is the right time to put together all architecture components, which were described in the previous sections of this paper, and provide its general overview. The following figure demonstrates it.

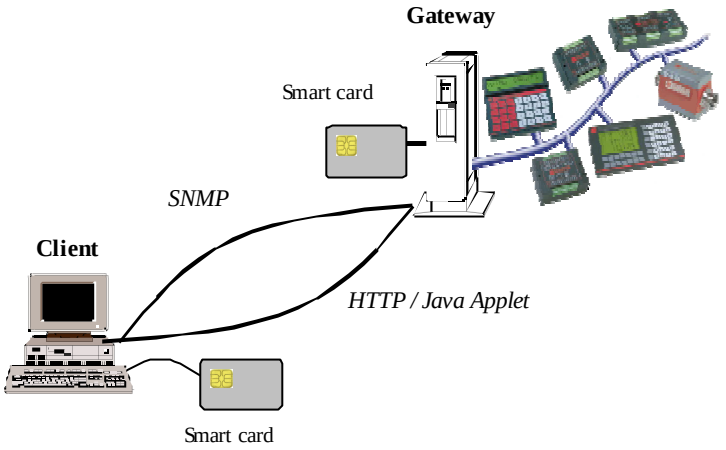

Figure 3: Secure architecture in general

A gateway plays a major role in the overall architecture. On one hand, it should provide a fully functional interface to the P-NET system. On another hand, it provides an access to it from the outside world. This could be mainly based on two groups of Internet protocols. One of them should be used to exchange data between a client and the gateway (therefore between the P-NET system). As we already wrote in this paper, Simple Network Management Protocol (SNMP) could be a perfect candidate to this role.

It might happen in some applications that we will need to provide system remote users a certain interface, which the can use to access the system from everywhere in the Internet. In the world of the Internet, it is quite difficult to image more universal user interface than an Internet browser. Internet browsers run on any platform, they are available at any PC connected to Internet. In this case we just download from the gateway WWW server a hypertext page together with a Java applet. This applet contains a whole client application, which is started in the browser and then interacts with the gateway. Of course, some application may not need such an interface at ll. In this case, it makes the overall architecture even more simpler.

A major aspect of the architecture is implementation of the data security mechanisms. They protect Internet data flows between the gateway and the client. Their implementation must be explicitly based on smart cards, which were already described in the previous sections of this paper. Otherwise it could be mostly impossible to guarantee their own security and therefore security of the whole system. In the following section of this paper, we will pay a more detail attention at their implementation.

Security concepts for the Internet gateway and client

As we already said above, it is also very important to apply certain security concepts to the Internet gateway and a client specification whilst developing them. This section of the paper will introduce the most essential ones. We will not go into implementation details because it is a rather huge topic, which could be subject for a separate paper.

The figure below demonstrates a general architecture of the gateway.

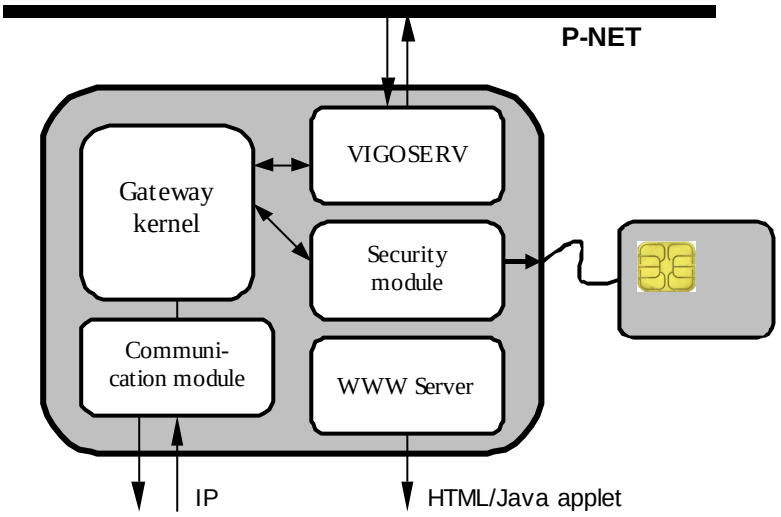

Figure 4: General architecture of the P-NET - Internet gateway

We would like to point out that it is very important to provide a secure communication to the smart card[6]. Usually a card reader (also called terminal) is connected to the serial port of a personal computer. It means that data sent to and received from the smart card can be easily disclosed and manipulated. It is one of the most critical points of the overall gateway architecture. The most reliable way could be to provide the security mechanically, i.e. putting the card together with the gateway hardware into one box, which can not be disassembled.

Simple Network Management Protocol framework was given an undivided attention in the previous sections of this paper. We have figured out that it can be a perfect basis for fieldbus – Internet applications, and therefore for P-NET Internet applications. The figure above represents main modules, which must be present in the gateway application in order to satisfy all requirements to it. All of them can be very easily expressed, and therefore implemented, in terms of SNMP. Gateway kernel incorporates a main part of the SNMP agent entity.

One of the most important issue on the kernel implementation, is a transparent communication to VIGOSERV from a Java environment. One of the approaches could be to use a "Bridge to ActiveX" Java Bean, which lets to integrate OLE object to a Java application [HuStei99].

[6] It is known as a "terminal problem"

The gateway security module implements SNMP Security and Access Control Subsystems, to security concepts of which we already gave an undivided attention. The security module is also capable of providing an access to security mechanisms running on the smart card. Taking into account a fact that the gateway software is implemented in Java, such an interface can be implemented in a really universal and flexible way. The communication module is not more than just an SNMP Dispatcher, which main task is to carry out a traffic management on the level of PDUs received from or sent to an IP port of the gateway.

As a conclusion, we would like to say few words about the client. Its implementation is also based on the concepts already introduced in this paper. As we already wrote, the most flexible form of its implementation is a Java applet, which actually consists of an SNMP manager application with a certain user interface. It also includes main SNMP entity modules including the Security Subsystem, which utilizes security mechanisms of the smart card. Java architecture provides additional security measure. The code of the applet can be digitally signed. This only improves the overall system security.

Conclusion

We think that security aspects start to play a dominant role in any fieldbus application, mostly in an application dealing with global networks like the Internet. A need in security mechanisms integrated to such applications is dictated by realities of today. History of computer systems development already has one example. The first prototype of the Internet actually covered only four universities in USA and was dedicated especially to academic purposes. Due to this fact, there were completely no reasons to consider any aspects of data security. However, it has started to grow up very rapidly. As a consequence, requirements to data security started to play more and more important role. It resulted in a big family of secure Internet protocols and services available today. No doubts, fieldbus technology will follow the same way.

Literature

[ISO89] ISO 7498-2, “Information processing systems - Open system interconnection – Basic Reference Model – Part 2: Security architecture”, ISO, 1989

[Sch96] Bruce Schneider, “Applied Cryptography”, John Wily & Sons, 1996

[Sta98] William Stallings, “SNMPv3: A Security Enhancement for SNMP”, IEEE Communications Surveys, Forth Quarter 1998, Vol.1, No. 1

[RaEf99] Rankl Wolgang & Effing Wolgabag, “Handbuch der Chipkarten”, 1999

[RFC2274] U. Blumenthal, B. Wijen, “User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)”, RFC 2274, January 1998

[RFC2275] B. Wijen, R. Presuhn, K. McCloghrie “View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)”, RFC 2275, January 1998

[Lemm98] Helmuth Lemme, "Wie sicher sind Chipkarten ?", Elektronik, vol. 16, 1998

[HuStei99] S. Hutter, R. Steiner, “Connecting a P-NET Fieldbus System to Remote operations via Internet”, Report on Computer technology laboratory work, ICT, TU Vienna, 1999