Security Considerations for FAN-Internet connections

Peter Palensky, Thilo Sauter
Vienna University of Technology, Institute of Computer Technology
Gusshausstrasse 27/E384
A-1040 Vienna, Austria
{palensky, sauter}@ict.tuwien.ac.at

Abstract

The interconnection between field area networks and IP-based LANs as well as the Internet as a whole is becoming more and more popular. Emerging security issues have been neglected or underrated in the past. However, traditional security concepts that work well for LANs and the Internet are hardly applicable to fieldbus systems. Based on an example from home automation, we review the problem and consider ways to prevent attacks both from the outside world and from within the fieldbus. Particular emphasis is given to firewalls, which are found to be of only limited value for securing fieldbus-Internet gateways. To tackle the security problem both on fieldbus level and in the Internet, we propose the use of smart cards for authentication and encryption. We discuss modifications that are necessary to make fieldbus nodes secure, why smart cards are "different" and strategies to implement access control on the gateway.

1. Introduction

In the recent past, the breakthrough of the Internet as a worldwide accepted communication medium has stimulated much work in the area of fieldbus/Internet connections. The benefits of a globally available Internet access to distant field area networks (FANs) are promising, as this both permits and facilitates important services like remote monitoring, control, or maintenance. Hence it is no wonder that FAN/Internet gateways are becoming more and more popular. With the still developing field of home automation, it can be foreseen that the application of such gateways will become even more attractive in the future.

There are many possibilities to realize the connection between a remote station in the Internet and a fieldbus. Since the gateway constitutes the only access point to the FAN and usually does more than just protocol conversion, we also use the well-known Internet term proxy for it. So far, most authors were concerned only with the feasibility of the proxy design. The emerging security problems were left aside. Still, opening a remote access to a fieldbus naturally raises the question of how to control this access. If we think about connecting a fieldbus to a network that can be reached by a large group of people, we must also tackle the security problem. Finding a way to prevent misuse is a prerequisite, the only alternative is to provide no connection at all and to leave the FAN isolated.

The severity of this problem clearly depends on the network the FAN is connected to. In many cases, FANs are connected just to IP-based LANs for process monitoring purposes. Such closed intranets provide a basic level of access control since they can be used only by a restricted group of people that can be controlled efficiently, so attacks from the outside are not to be expected. On the other hand, it is well known that most successful attacks in companies come from within the system and not from the outside [1]. Consequently, there is also a need of securing a FAN proxy in an otherwise closed environment.

Things are a bit different in home automation. Here the predominant wish is to control household appliances from the user's office desk, which makes an Internet connection indispensable. The prevailing threat in this case is certainly an intruder from the Internet gaining control over the home network. The potential damage is to be seen rather in vandalism than in the loss or espionage of critical data. On the other hand, the probability of attacks from the inside is negligible, although there are situations conceivable where exactly this possibility is a particular challenge for the implementation of security mechanisms. We shall discuss these cases and possible solutions in the following sections.

2. Security theory for FAN/Internet proxies

Talking about fieldbus-Internet connections, we understand that the connection is achieved by a single gateway or proxy agent. This gateway provides a suitable protocol conversion between the Internet (or equally an IP-based LAN) and a specific fieldbus protocol (Fig. 1). In many cases, this proxy will also have caching mechanisms for a set of data objects in order to facilitate data exchange with clients in the outside world. Such a structure has been found useful from both performance and implementation points of view [2].

WFCS-2000, September 6-8, Porto, Portugal
0-7803-6500-3/00/$10.00 @2000 IEEE

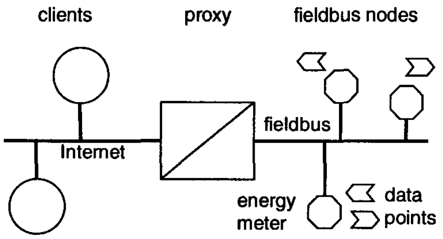

Fig. 1: FAN-Internet connection via a single gateway

When we talk about security issues, we are not only concerned about the Internet part of the interconnection. Security also comprises the fieldbus with its nodes. To better convey our arguments, we shall discuss several points by means of a simple example. Consider a home automation network connected to the Internet via a gateway. One node of the FAN is an energy meter that can be monitored and controlled for instance by a utility company. Other nodes are light switches or a heating control that can be accessed by and maybe only by the owner of the home. This exemplary setup is complicated enough to demonstrate the fundamental security problems a useful Internet connection may entail.

2.1 Security considerations for the Internet part

Let us begin with some basic considerations. A gateway or a proxy that connects a network of fieldbus nodes to the Internet has to provide the following security mechanisms:

1. authentication of accessing Internet clients
2. privacy for all transmitted data on the Internet
3. access control for the managed data points

In the Internet it is pretty easy for an intruder to fake IP packets and to perform IP masquerading, i.e. pretending to be some other user with a different IP address. To prevent this, additional means of authentication have to be used. Authentication is usually done on different levels. User name and password are required to get access to the Internet at all, and the FAN gateway or some server can issue specific certificates for authentication. The particular problem is that all passwords and certificates have to be distributed somehow. If this is done via an a priori insecure channel like the Internet itself, the whole authentication cannot be trusted anymore. Electronic banking is sometimes done by using Transaction Numbers (TANs) that are printed on paper and distributed via conventional post to the clients. This is of course far too complicated and inflexible for remote FAN access. A possible alternative are smart cards that can help to overcome the problem of key or certificate distribution as will be shown later.

Privacy is maybe the best known aspect of security. The data have to be encrypted and signed so that it cannot be read or altered by an unauthorized party. This can be done via some public cryptographic algorithm that uses either symmetric or asymmetric keys. The art of privacy is then to store and distribute the keys in a secure and safe way, while the algorithm is public and known by everybody. The method of applying the algorithms is not the problem, the more important part of privacy and security in general is key management and the security policy.

The third aspect of proxy security in the context of FAN access is access control. A FAN node consists of a set of communication objects, we call them data points in the sequel. Every FAN type has its own interoperability association defining standards on how these data points should look like, how they are encoded and how they are addressed. The gateway represents a number of nodes and/or a number of such data points. These data points can be either FAN-wide shared and distributed variables or just parts of an internal memory on the fieldbus nodes that may be accessed via the fieldbus. The common property is that some data points can be only read, while others can be written as well. Whether or not write access is possible at all is usually checked by the node itself. The gateway, on the other hand, has a global view of the FAN. It connects a number of users to a number of data points and has to keep track of which user may access which data point or FAN service in which manner. This is usually done with an access control matrix described later in this paper.

There are other aspects of a secure system like “denial of service”, “availability” and other “qualities of service” that play an important role [3]. If a switching command is sent to a device somewhere, it is often required to have the guarantee that the command is received within a certain time and that the receiver cannot deny that it has received the command. Unfortunately, the current state of the Internet does not allow such “qualities” or functionalities. There are no bandwidth guarantees and if dial-in lines of an ISP are used for the Internet connection, the availability of a free line cannot be guaranteed either.

The aforementioned aspects are certainly the most important and typical ones for a gateway. But the Internet offers other threats as well. One of them is to be overloaded with dummy-requests. If a gateway is present in the Internet, it can be kept busy with refusing access of a user that attempts access repeatedly within short periods. IP-based networks offer some means to deny routing between specified domains, but this limits the flexibility and usability of the gateway. It should be accessible from everywhere in the world without complicated methods. As it stands, IP is not able to supply all necessary means of security and reliability for FAN-Internet proxies, but it can be used to fulfil at least the basic functions of a secure FAN access.

2.2 General Security Problems of FANs

The other side of the medal is security on the field level. It is of course possible to implement security mechanisms on each node, sometimes parts of it already exist (like authentication in the case of LonWorks [4]). But usually the standards and specifications of field area networks would be violated if one tried to introduce particular security mechanisms. The easiest way is to use the fieldbus only as a transport channel and to tunnel secure packets over standard FAN protocols and services (see section 4.1). However, adding such a security mechanism is likely to cause problems with interoperability.

Still security may be necessary also on the fieldbus itself. In our example, the utility company has a vital interest to prevent the user from manipulating the setup of the energy meter via the FAN. This is even more important because the owner of the home can easily gain physical access to the fieldbus. Consequently, appropriate mechanisms must preclude an intrusion into the fieldbus node via the fieldbus itself. Maybe the user is allowed to read out measurement values. But he certainly must not change the configuration of the device.

It is easy to encrypt and sign measured data and represent it in a data point. The problem is that all other nodes do not understand this encoded data point. One possibility to overcome this problem is to duplicate the data points for secure access.

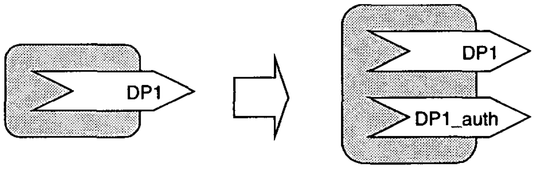

Fig. 2: A FAN node is extended by an authenticated and signed data point

This makes sense only for authentication and signing of data points. To ensure privacy, it is no good idea to have a data point both encrypted and unencrypted at the same time. The right part of Fig. 2 shows a FAN node with two data points. One complies with the corresponding FAN interoperability standards, while the second one implements some proprietary authentication and signature mechanism. The purpose of the signature is to protect the data point against unnoticed alteration. DP1_auth thus contains the actual value of the data point, its signature, and the authentication of the node. A node that wants to read DP1_auth needs to use the same algorithms and the correct keys to verify the authentication and the signature. This node is either the gateway or any other node that belongs to the same non-standard-conforming subsystem of the FAN.

If a data point is confidential, it has to be encrypted as well (Fig. 3). It is clear that the encrypted data points can only be used by the proprietary subsystem. A signed data point, on the other hand, contains still the data in their original format, which can be understood by conventional FAN nodes even if they disregard the signature itself.

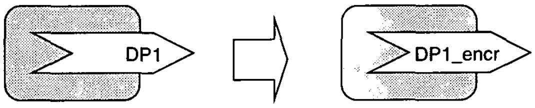

Fig. 3: A data point is encrypted

But the problem of secure data points goes deeper. If we regard how such an abstract data point is formed in reality, we recognize that the FAN protocol is not the only thing that has to be made secure. Before the data can actually be transmitted over the fieldbus, a physical quantity must be transformed by a sensor into an electrical signal that is processed by the fieldbus node (Fig. 4). In this local information flow, there are several possibilities to gain access to the data even before they can be encrypted. Consequently, precautions should be taken for the entire information source. The usual way is to encapsulate the source of information, the sensor and the node in a closed and secure containment as it is done with energy counters. But even if this is feasible, it is still not guaranteed that the node itself is secure. Sometimes the firmware and the application of nodes are downloadable, and if not, it is often possible to bypass the running software by special network management commands like reading out the memory of the node (that contains the raw data of the sensor, the keys, etc.) via the network.

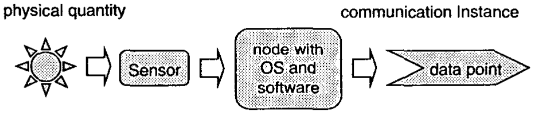

Fig. 4: Information flow from the real world to the data point

Finally, a secure node does not necessarily lead to a secure system. Even if the data point and its underlying parts are secure by themselves, it is still possible to attack the system at other points like the gateway, the Internet, or the client machine. Particularly bad is the situation if we use open media such as wireless transmission channels. Then it is, in principle, even possible to access the FAN without a physical connection to the wires or without entering the facility where the FAN is installed. Almost all parts of the system shown in Fig. 1 are insecure and unreliable. Taking a somewhat paranoic view, we must be afraid of many potential dangers: other FAN nodes might contain a "trojan horse", the Internet is a playground for intruders, and every possible operating system or application program used for the gateway and the client must be considered to be insecure. In fact, the entire way of the information from the source to the sink has to be made secure including the source and the sink.

Returning to our example of energy counters that are operated and read by a utility company via a fieldbus and the Internet, we conclude that the values have to be encrypted and authenticated to prevent faked data and to protect the privacy of the customers. The question is where to apply which means of security. Therefore we have to take a look at the functional parts of such gateways and some security technologies.

3. Realizations and security technology

3.1 Gateways and Proxies

The Internet-view of a general FAN/Internet proxy consists of several components or objects that are responsible for access control or management (Fig. 5).

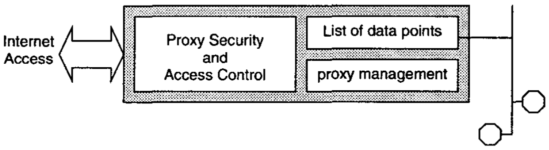

Fig. 5: Parts of a FAN/Internet proxy

The Management component is used to configure the proxy. Parts of this configuration are FAN-independent like IP address and telephone number of the ISP, error statistics, etc. Other parts are FAN-dependent like timeouts, driver parameters, FAN-specific addressing-data and the like. The proxy security component is responsible for managing the keys used for encryption and authentication and for storing the registered users and their rights for the FAN system.

The list of data points represents the proxy itself. Every entry in this list contains data of one discovered or specified data point such as its value, its name, its type, a description what this data point is and where it is physically located, maybe a timestamp of the last value or any other information that such a data point offers.

It does not matter, whether these components of the proxy are represented by separately addressable programs, by some SNMP MIB branches, HTML-pages or CORBA objects. The meanings and objectives of these objects are always the same, simply the addressing and the data representation change with the technology used for the proxy.

3.2 Firewalls

The most widely employed security measure in the design of network interconnections today is the firewall. Indeed the term "firewall" is often used as a synonym for network security, and we thus could suspect that a firewall placed somewhere between our fieldbus proxy and the Internet should provide enough protection. But is this really so?

The main purpose of a firewall is to separate a private, restricted intranet from a network that is accessible for a larger community [5]. This separation is normally implemented on IP-level by granting access only to computers from a predefined set of IP addresses or domains. Only requests from such addresses are accepted by the firewall, all others are denied. This requires that the firewall also works as an IP router connecting the private IP segment and the "open" segment reachable from the Internet. As the entire network traffic is controlled by the firewall, the intranet behind it is completely hidden from the outside world. For the utmost security level, the IP addresses used in the intranet are not even made visible to the outside. Therefore, we can use arbitrary addresses in the intranet with the special case of a "masquerading" firewall. The other way, from the intranet to the Internet, is open: the firewall directly forwards the request from the internal node (the FAN proxy in Fig. 6). The server in the internet sends the response to the firewall, who in turn forwards it to the requester. Ideally, the node in the intranet does not even notice the presence of the firewall. It is evident that this forwarding mechanisms require the use of connection-oriented protocols like TCP. With connectionless protocols like UDP, the firewall cannot remember the originator of the request and therefore cannot forward the response.

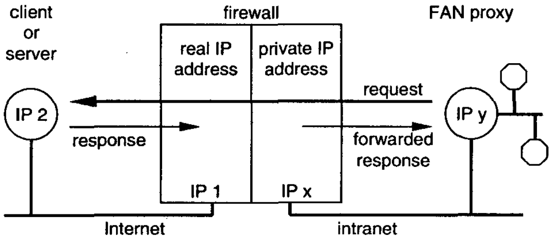

Fig. 6: Masquerading firewall with the FAN proxy located in the intranet and permitted access directions.

The problem with such a masquerading firewall is that it prohibits the direct addressing of a node in the intranet. The FAN proxy, on the other hand, provides some sort of server functionality and must therefore be addressable. Now if a client from the Internet wants to access the proxy, it cannot do so unless the IP address of the proxy is made explicitly known to the outside world, which is often undesirable for security reasons. The situation is improved if the proxy is located in front of the firewall, as shown in Fig. 7. Here the client in the intranet can easily reach the proxy, and the firewall forwards the respective response.

This approach has two deficiencies. First and worst, the proxy is no longer protected because it is part of the "open" network. Second, the proxy is not able to contact the client. From the viewpoint of client/server functionality, this is not necessary anyhow. But for a fieldbus proxy, this is a vital function since the proxy must autonomously monitor alarm conditions and generate alarm messages if required. With the addressee of such messages being unreachable behind a firewall, this function is severely impaired.

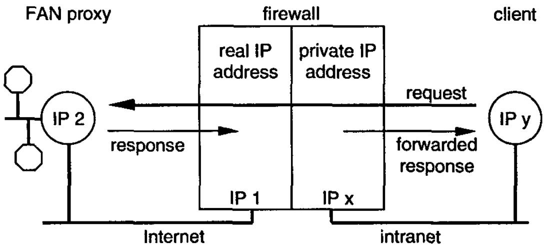

Fig. 7: Masquerading firewall with proxy outside the intranet

The aforementioned considerations suggest that a firewall is completely unsuitable to protect a fieldbus proxy. Yet there is a way out of the dilemma of mutually restricted accessibility. Actually it is a workaround to overcome the visibility problem of the proxy from the outside world. The proposed structure is to integrate the proxy and the firewall on the same machine. In this case, the IP address of the proxy is also the address of the firewall. Still the proxy application cannot be accessed directly. All requests from the Internet are directed to the firewall, and connections are granted only to known IP addresses or domains. The firewall forwards honored requests to dedicated IP ports that are observed by the proxy. Likewise, the proxy returns responses (or notifications) to predefined ports of the firewall. This way, the intranet is not needed at all, if the proxy is the only node protected by the firewall.

Of course it is possible to have firewall and proxy run on different machines and to use port forwarding. From the point of functionality, these two strategies are fully equivalent. However, the use of the intranet entails an additional communication overhead, which reduces the performance. The integration of proxy and firewall is thus much more efficient.

The simplest configuration completely dispenses with the intranet and uses a dedicated firewall for the fieldbus proxy. Often, however, there are also ordinary IP nodes to be protected, and two firewalls are too expensive in terms of maintenance. A typical example are small networks like in home automation, where we need a FAN access and at the same time have a few computers that need access to the Internet. This case can also be tackled by the proposed approach. The only difference is that we now must distinguish between connections intended for fieldbus access and those targeted at the intranet. As we have only one IP address, the best way to separate these two access types is to use different ports of the firewall's protocol stack. For the sake of practicability, the access to the intranet should then employ the well-known ports, whereas the access to the FAN proxy could use dedicated ports that must be known to the machine establishing the connection. Fig. 8 shows the structure of this combination of firewall and proxy.

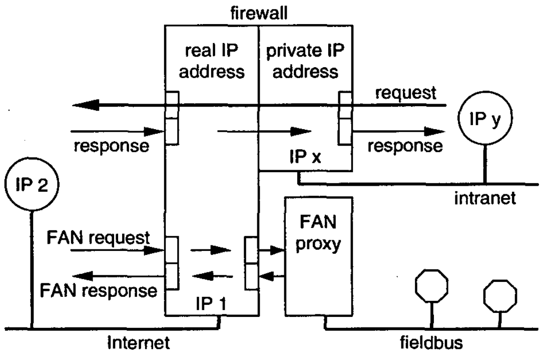

Fig. 8: Proposed integration of FAN proxy and firewall with port forwarding and an additional intranet

The use of different ports for intranet and fieldbus access has the additional benefit that the connection restrictions can be configured independently for each port. Another particularly convenient advantage is that the restriction concerning connectionless protocols mentioned earlier no longer applies to the proxy (it still does to the intranet!). Since dedicated ports are employed to contact the fieldbus proxy, it is in principle possible to use UDP to exchange data from the proxy to some machine in the Internet. The firewall can still handle UDP-based responses correctly because the proxy is the only entity associated with these ports, and there is no uncertainty where to forward the data packets.

3.3 Security protocols

Firewalls inhibit or allow access to certain nodes or ports of the protocol stack based on an analysis of the sender's IP address. While this is a basic level of access control, it is not enough to provide security. IP addresses can be faked, and the data transmission itself is still insecure. So, in order to protect also the data flow, additional measures need to be taken. Between the firewall and the client in the Internet, secure protocols are indispensable.

In the Internet, several protocols are used that provide security extensions. Some of them, like SSL (secure socket layer) are only intermediate layers placed between TCP/IP and application protocols [6]. Others, like S-HTTP, APOP, or SSH (secure shell), have security features directly included in the application protocol. For network management purposes, the latest version of SNMP also includes security mechanisms [7]. A promising approach is IPsec [6], which is a secure version of IP and is frequently used for setting up virtual private networks.

All these protocols are proven and widely used. However, they require at least an IP connection. That is, they are restricted to the Internet or an IP-based LAN. They are applicable for securing the connection between the client and the FAN/Internet, but they cannot be used to establish a secure connection over the fieldbus, unless they are tunneled over the FAN. This is a not very feasible approach, because it would mean to implement IP and higher layer protocols on the fieldbus nodes, which may be resource-consuming.

Even if secure protocols are being used between the client and the proxy, a second problem remains: The keys for the encryption and authentication mechanisms must be given to the communication partners. Key distribution is known to be a rather intricate matter [8] as one has to start with an insecure connection. In many practical cases, keys are therefore transmitted not via the Internet, but by post or other non-electronic means. This involves someone typing the key into a configuration program, which is not user-friendly and error-prone. To overcome both drawbacks, the key distribution as well as the security problem on FANs, we propose to use smart cards instead, as will be outlined in the following.

4. The smart card approach

Smart cards, also known as processor cards, play an important role in our understanding of a secure computer system. Any security-relevant application or data should be placed on a smart card, because they are at present the only secure containment for computer programs and their data at a reasonable price.

Smart cards are usually plastic cards that contain a microchip and external contact pads. The interface to the chip is an asynchronous serial interface. Most cards use a standard smart card protocol like T0 or T1 [9]. The internals of the chip are a processor (like an 8051) and both volatile and non-volatile memory (some kbyte). A smart card contains an operating system and applications like any other micro controller system [10].

But why can we trust smart cards when we cannot rely on ordinary micro controllers? The reason is the design of these cards. The only way to access them is via the serial interface. There is no accessible bus between the processor and the memory. It is almost impossible to bypass the interface program on the card. Smart cards are designed to protect a certain value by making it extremely costly (i.e. more than the protected value) to crack them and get this value. This principle is used for electronic money and other forms of electronic commerce.

The most important feature of a smart card in our application is that it can generate a pair of keys and keep the private key on the card. Not even the programmer or owner of the card knows the key or can get access to it. Furthermore, it never leaves the card, because it is used only on this card by the programs and algorithms on the card. This makes it both user-friendly and secure. Besides, smart cards encapsulate the different programs on the card that no program can attack or analyze another program.

These and other reasons make a smart card an inexpensive and secure device. If, in the applications considered here, the gateway has to sign and encrypt a packet, it gives the original packet to the smart card, which then processes it. The ready packet is delivered back to the gateway and can be sent to a receiver that has the corresponding smart card to decrypt and verify it. There has always to be one smart card on either side of a communication channel to enable real security. In fact, from the viewpoint of security, the communication takes place directly between the smart cards, and all other network entities see nothing but encrypted and thus senseless data packets.

Smart Cards can generate, protect and use their keys in a way that inhibits almost any attack. But the information that has to be made secure by the smart card is usually not generated on this card. It comes from a source of information outside the card, as we have shown in section 2.2. At this point, the security-relevant data are vulnerable. This means that using smart cards does not fully solve any security problem, it has also to be ensured that the data is brought to the smart card in a secure manner. The communication afterwards between two smart cards is supposed to be secure, but on the client side the same considerations have to be made when the data leaves the receiving card again.

4.1 The secure FAN node

Securing a fieldbus node is much more than just equipping it with a smart card. To protect the node against attacks from the FAN, additional precautions are necessary. Let us hinge these considerations on the energy meter example. A first attempt to introduce a smart card into the node is shown in Fig. 9. This somewhat naive approach just extends the existing configuration comprising the actual sensor and the FAN controller by the security device. The basic idea is that in order to make the entire node cost effective, the smart card interface can be handled by the controller itself (provided it as enough spare computing power). The sensor data are then fetched like in the insecure version of the node and transferred to the smart card, where it is encrypted. Afterwards, the now secured data are transmitted over the fieldbus.

Although this idea sounds quite reasonable at first, it has a severe security hole. In the course of the processing of the sensor data, they must be stored in the memory of the controller before being transferred to the smart card. Consequently, they are available in plain text at least for some time. The trouble with this procedure is that in many fieldbus systems, the RAM of the controllers must be accessible for configuration purposes through network management commands. Therefore, an intruder having access to the fieldbus (which is no problem at all in home automation) may simply read out the contents of the controller memory. Of course it takes some knowledge about the internal structure of the controller and some efforts so guess what the data actually mean, but in principle it is possible to read or – even worse – modify the data before they are being sent via the smart card. This potential risk shows that simply attaching the smart card to the FAN controller is an inappropriate strategy to realize a secure node unless direct memory access via network management commands can be unconditionally restricted.

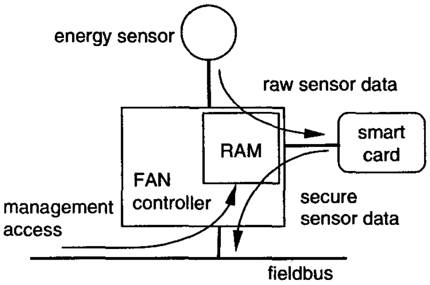

Fig. 9: Direct management access to raw sensor data despite presence of smart card

It is unrealistic to increase the security of a node by completely turning off management access, since this would in turn be a great hindrance for the configuration of the FAN. We propose a different solution, where the raw sensor data are processed by the smart card *before* being read by the FAN controller. This way, the controller never stores the raw data, and an intruder can at most gain access to encrypted – and therefore useless – values. All that is needed to realize this procedure is an additional (non-manageable) controller between the sensor interface and the FAN controller as shown in Fig. 10.

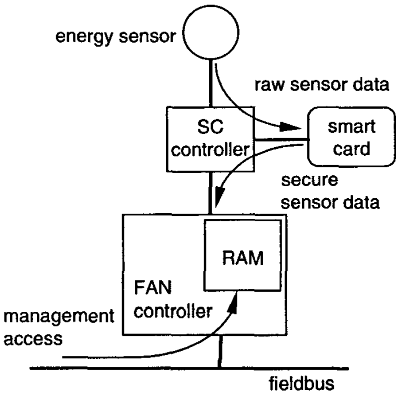

Fig. 10: Proposed secure architecture with additional smart card controller

The smart card controller can either be an appropriate microcontroller or better a dedicated ASIC acting like an intelligent switch. At any rate, it must thoroughly separate the data flows between the sensor, the smart card, and the FAN controller. In case of a controller solution, it must have separated memory areas for the raw and the encrypted data, the former being inaccessible for the FAN controller. On the other hand, it must also provide a communication channel between the fieldbus and the chipcard for authentication purposes. The proposed architecture still permits configuration and management of the fieldbus node with standard tools, but at the same time efficiently restricts the access to the critical sensor data. The price to be paid is an additional device.

It is important to notice that a secure fieldbus node is secure only as long as it is physically intact. An intruder must not be able to gain access to the circuitry inside the node. Ensuring this is mainly a question of mechanical construction and stability. In particular, the housing should be properly sealed and perhaps equipped with sensors that can detect attempts to disassemble the node.

4.2 Implementation of access control

Access control for a proxy or a management agent that controls a number of objects is often done via a matrix containing specific user rights. Registered and authenticated users are granted access rights for registered managed objects like the data points in our FAN-Internet Gateway. The relation between users and the data points can either be read/write, read, or not accessible.

An example for sophisticated user rights is a high-tech energy meter that performs the counting of consumed energy as well as demand-side-management (DSM) in terms of electrical energy. Table 1 shows a possible matrix with the respective user rights. We have three users: the utility company (Utility), the owner of the facility (Owner), and the DSM provider (DSM). The node hosts three data points that are relevant for the users: the measured total energy consumption (DpEnergy), the DSM-configuration (DpDsmConf), and the configuration of the energy counter (DpCounterConf).

Table 1: User rights for managed data points on a networked energy meter

	Utility	*Owner*	*DSM*
DpEnergy	R	R	R
DpDsmConf	-	-	RW
DpCounterConf	RW	-	R

The dimensions of this matrix change with the number of users and the number of accessible objects. Therefore the matrix is often represented as a table with three rows: {object; user; rights}. Together with this table, other tables contain the actual users, the objects and their properties. These tables have to be stored in a secure place. The question is if this is done in a centralized way by the gateway or in a more distributed fashion. Let us suppose the tables are stored on smart cards. The two basic possibilities are:

1. each managed (physical) object (data point or node with a number of data points) has its own smart card containing the users of this object and their rights,
2. the gateway maintains all necessary tables on its smart card.

The first option is a quite lavish one. Each data point or each node has to be equipped with a smart card and a smart card reader (see Fig. 11) and is directly accessible from a client in the outside world. The gateway simply passes the encrypted data from the FAN nodes to the clients and vice versa.

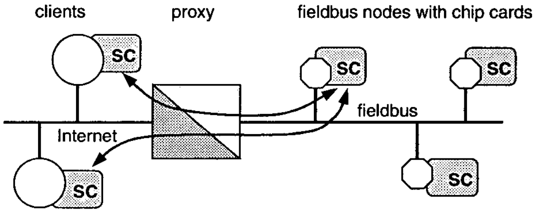

Fig. 11: Decentralized access control

This results in a decentralized way of access control. It is the most flexible and secure way to handle this problem but also the most complicated one, since new users or access rights must be updated on the respective smart cards. To enable authentication of different users, a proprietary protocol must be placed on top of the standard FAN application layer. This way of bringing the functionality of access control into the field level would violate FAN standards and agreements on interoperability. In general, user accounting and access rights of networks are maintained in a centralized way.

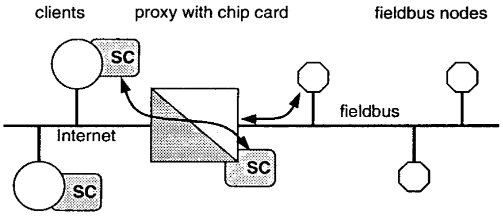

Fig. 12: Centralized access control by the gateway

When the gateway is the only one that performs access control (Fig. 12), the managed network itself remains open, and any locally connected user can access data points. This means, the FAN and its data has to be protected physically from any kinds of attacks in terms of unauthorized access. In the light of our energy meter example, this alternative is undesirable for obvious reasons.

A compromise would be to equip all involved communication partners with smart cards (Fig. 13). The authentication between the nodes and the gateway would then be point-to-point, there is no need to maintain and configure multiple users on the smart cards of the nodes because there is only one user for them – the gateway itself. The smart cards at the node side can then be small and cheap, because they have to store only the relevant data for one user. In this case, the data flow is disrupted by the gateway, which acts as a mediator. The access of the DSM company to the energy meter data would thus roughly follow a two-step procedure. The DSM client logs into the gateway and demands access to the measured data. The gateway in turn logs into the respective fieldbus node and fetches the latest values. These data are encrypted by the node's smart card and passed to the smart card of the gateway. On the card, the data are decrypted and encrypted anew for the communication with the DSM client. Finally, the gateway sends the data to the client. It is important to note that the processing by the gateway does not cause security problems, because the processing is entirely done on the smart card. Plain text data are nowhere accessible on the gateway itself.

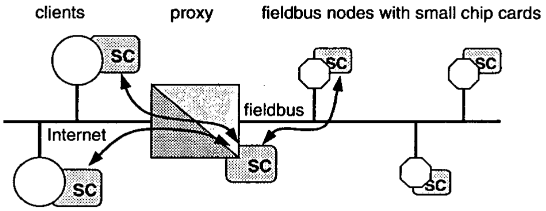

Fig. 13: Simple smart cards at the nodes

Generally we can say, that the simpler way to achieve security for the Internet access is to secure the gateway. But although this possibility makes maintenance easier, it still has a big drawback. Smart cards are not intended to be "updated" via some network. It is usually not wanted that a smart card can be written via a network, so we cannot manage these tables if they are stored on the card. New users that have to be added or user rights that have to be changed lead to a new smart card that has to be programmed, sent via post and replaced at the customer's site.

To overcome this, the user rights can also be stored on a rewriteable media on the gateway like a flash memory or a hard disc drive. The usage and maintenance of these data still has to pass the smart card on the gateway for security reasons.

5. Conclusions and future work

The connection of fieldbus systems and open networks like the Internet is useful for various reasons, but raises security questions that have been neglected in the past. We have shown that it is not enough to prohibit unauthorized access from the outside world to the FAN gateway. The security on the fieldbus level is almost as important. In particular, attacks from inside the FAN must be avoided by appropriate measures. This includes both software-based strategies like authentication and encryption as well as a physical protection of critical and security-relevant parts of the system. Firewalls can be used in combination with other strategies, but require a very careful setup and are by no means sufficient to provide real security.

The implementation of security mechanisms is difficult on standard fieldbus nodes. Not only is the available computing power usually limited, some fieldbus protocols also permit a direct access to the memory of the nodes via network management commands. In our opinion, smart cards are a suitable way to tackle this problem. They efficiently encapsulate encryption mechanisms and can be used to establish a secure communication channel. Unfortunately, smart cards are not yet well established. There are many different products, but no real standard, let alone a standardized interface protocol. However, their increasing employment in electronic banking might lead to one or a few standards in the long run.

One must be aware of the fact that an improved security requires a fair amount of accompanying organizational measures even when smart cards are used. The configuration and management of secure systems is a crucial point, in particular in an industrial environment. The distribution of the cards is only one logistical problem. Still worse is the handling of error conditions. What happens if a card becomes defective and fails? Should the affected node shut down its operation completely or rather enter a failsilent state? Coming back one more time to our example, we deem it highly undesirable that the energy meter interrupts the whole energy supply just because the fieldbus node is no longer reachable in a secure way. Strategies to cope with faulty or misbehaving nodes play an important role in a secure network. The only thing that does seem clear is that no backdoors should be allowed to reanimate nodes with faulty smart cards via the fieldbus. Even though is might be convenient for maintenance and the upkeeping of a continuous operation, it opens a tempting security hole. Provided we chose appropriate strategies, a secure connection between FANs and publicly accessible networks seems feasible. Organizational questions, however, remain.

References

[1] R. Roger and S. Meyers (Eds.), *Maximum Security*, Sams.net Publishing, 1997.

[2] M. Knizak, M. Kunes, M. Manninger and T. Sauter, "Applying internet management standards to fieldbus systems", *Proceedings of the 1997 IEEE International Workshop on Factory Communication Systems, WFCS'97*, Barcelona, 1.-3. Oct. 1997, pp. 309-315.

[3] D. B. Parker, *Fighting Computer Crime*, John Wiley & Sons Inc., 1998.

[4] *Motorola LonWorks Technology Device Databook*, Rev. 4, Motorola Inc. 1997.

[5] S. Strobel, *Firewalls für das Netz der Netze*, dpunkt.verlag, Heidelberg, 1997.

[6] M. Raepple: "Sicherheitskonzepte für das Internet", dpunkt.verlag, Heidelberg, 1998.

[7] W. Stallings, *SNMP, SNMPv2, SNMPv3, and RMON 1 and 2*, Addison Wesley, 3rd ed., 1999.

[8] W. Funny, "Key Management Techniques", in B. Preneel and V. Rijmen (Eds.): *COSIC'97 course*, Springer Verlag Berlin Heidelberg, 1998.

[9] W. Rankl and W. Effing, *Handbuch der Chipkarten*, Hanser, 1999.

[10] R. Bright, *Smart Cards: Principles, Practice, Application*, Ellis Horwood Ltd., 1988.