SECURITY IN GLOBALLY DISTRIBUTED INDUSTRIAL INFORMATION SYSTEMS

Petri Saloma, Ronja Addams-Moring, and Teemupekka Virtanen
Telecommunications and Multimedia Laboratory, Helsinki University of Technology

Abstract Today's industrial market is global. Manufacturing and measuring products are geographically distributed all over the world. Furthermore, the products are sophisticated and include know-how, often in the form of software. Thus the manufacturer is often the only party that can provide the maintenance services. In this paper, we focus on single intelligent and resource limited devices, which need to be remotely monitored or controlled. With the global Internet, satellite and telephone networks the access itself becomes possible. However, security is a challenge, because the remote devices' limited resources are pitted against threats from the Internet. We discuss the assets of and threats against a globally distributed industrial information system. To protect the assets, three possible security architectures, Centralized connections, Layered architecture and Integration of centralized and layered architecture, are proposed. Of these three, the integrated architecture meets best the requirements of a globally distributed industrial information system.

Keywords: Network security, Security policy, Information flow

1. INTRODUCTION

Nowadays, the manufacturer of a car is the only one, who has the know-how needed to repair any non-trivial problems in the car. The situation in industry is becoming quite similar – only the manufacturer of a device has the know-how needed for its maintenance. The devices are getting more sophisticated and they are distributed globally. Furthermore, industrial companies are themselves global and cooperate with other companies (suppliers of devices, maintenance services, Internet Service Providers). (figure 1)

The device manufacturers must often provide maintenance services remotely, because repair personnel cannot travel to every single device sold worldwide. Further, the device's remote control and monitoring capabilities might be valuable for the device owner.

Y. Deswarte et al. (eds.), *Security and Protection in Information Processing Systems*

The focus of this paper is firstly on single devices which are crucial parts of the industrial systems they belong to and which need to be remotely controlled or monitored. Such remote control and monitoring has become possible through the development in Information Technology (figure 1).

In this paper we aim to find one or more useful secure architecture concepts for how integrated single devices could be remotely monitored and/or controlled. We study three alternatives: the centralized architecture, the multi-zoned architecture, and last, we conceptually combine the two into a hybrid architecture.

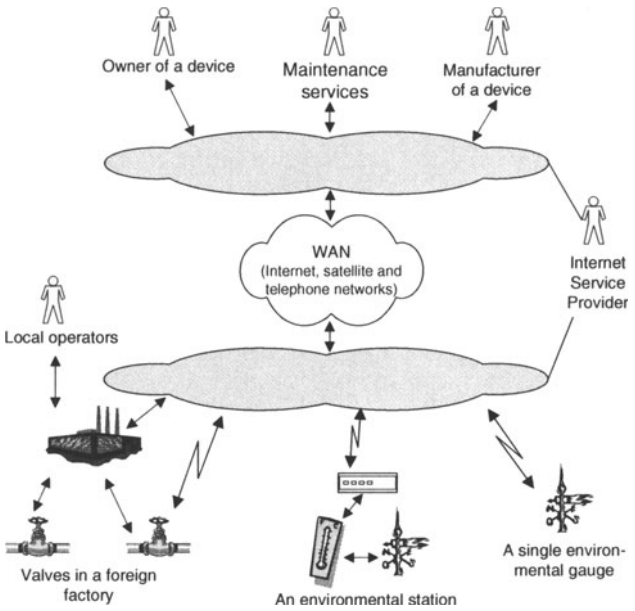

Figure 1. Parties in a globally distributed industrial information system, GDIIS

2. ASSETS AND THREATS IN A GDIIS

Because remote devices are connected to an open and insecure network, the security and trustworthiness of a GDIIS are imperative. From the device owner's viewpoint, information and services are the main assets of the system. Services can be divided into plain information query (e.g. a product recipe), monitoring (e.g. process temperatures) and controlling services (e.g. parameter setting or direct control of the device).

The remote device itself as well as the process and the system which the remote device is part of must be protected. Compromising even one of those

can lead to process or product flaws, to the device breaking or even to a plant explosion.

From the manufacturer's viewpoint, the software can be more valuable than the device itself. Hence, the software should never end up in a competitor's hands. It might also be necessary to update the software remotely.

Security threats against these assets are basically the same as in any system which is connected to an open and insecure network. Eavesdropping (including man-in-the-middle attacks), spoofing and Denial of Service (DOS) attacks are all such network related threats that might lead to the compromising of the remote device's security. The assets can be considered secure as long as the confidentiality, integrity and availability of the GDIIS are preserved.

Confidentiality

Most often, the information and software of a GDIIS is confidential (e.g. a product recipe). But, this may not always be the case. The remote device's data might provide no information without the know-how to interpret the values (e.g. configuration parameters) or have only little value for the attacker (e.g. process temperature).

Integrity

Generally, the integrity of information, services and software must always be ensured in the GDIIS. This holds especially for software and services where every bit is significant.

However, some level of integrity compromisation can be acceptable, because e.g. monitoring information often has much redundancy and is not necessarily security critical. However, this does not always hold even in monitoring data, because sometimes every change needs to be traceable and non-repudiable. One example is the U.S. National Lightning Detection Network (NLDN) used to locate lightning strikes (Vaisala,). The NLDN is used by insurance companies to expose insurance fraud.

Also, laws or authority regulations may set strict bounds for the integrity of information, e.g. US. Food and Drug Administration (FDA) has regulations for food and drug manufacturing practices (including information systems and information traceability). Hence, the software and services of the GDIIS must always be malware free, whereas some integrity flaws might be accepted in the information.

Availability

Services and information are often time dependent, and hence availability is emphasized. There is no point in a service which is not available, or in information which cannot be accessed. A no-access signal could also indicate

a broken device and a costly serviceman might be sent to the station without real reason. Therefore, even services in an environmental monitoring station must always be accessible when needed.

QoS (Quality of Service) is not a core security property, but has some security related features and can be included in the availability category. In industry, it is important to be able to predict the behavior of the system and instead of asking "How high QoS level on average can be provided?", we ask "Which QoS level can be guaranteed?".

3. SECURITY MECHANISMS

Security mechanisms are needed to protect the assets. In addition to technical security mechanisms, physical and psycho-social security mechanisms should be taken into account as well. Examples of physical security mechanisms are passage control and physical separation of a web server and the company's intranet. Psycho-social security mechanisms try to solve the problems associated with human beings, e.g. own personnel. Also, laws and regulations are psycho-social security mechanisms and they try to point out to the attacker that the benefits gained are less than the possible costs (consequences) (Naedele,).

General security guidelines

A system which has very little security is typically in great danger of being compromised. In contrast, a system with excessive security can become impossible to use. Therefore, at the minimum, the risks of the system must be evaluated. Further on, the pros and cons of used security mechanisms must be evaluated, and when the disadvantages are greater than the benefits, the mechanism should be left out.

Security might seem to have only a minor role in a remote device, because it does not directly hold any confidential components. However, even then the risks must be evaluated, because firstly, the components might include a backdoor to the company's intranet and secondly, sooner or later new components are added to the system or the existing components are updated. The integration of those new or updated components can lead to a security vulnerability, even if the new components are as secure alone as the existing system.

Authentication and authorization

Naedle states that also the connection itself (and not only the user) should be authenticated, and that authorization can be seen as action authentication. Hence, authentication can be divided into three: 1) *connection authentication* i.e. "Is this user permitted to connect the remote device at all?", 2) *user authentication* i.e. "Is this user permitted to use the remote application at all?"

and 3) *action authentication* i.e. "Is this user permitted to execute the action in question?" (Naedele,).

Connection authentication. Connection authentication specifies who in general is allowed to make a connection, i.e. to continue to the user authentication step. In TCP/IP networks, connections only from/to certain IP-addresses or ports can be permitted. Also, only certain applications might have permission to access the network / to be used over the network. (Naedele,) However, the spoofing of an IP address is quite easy and hence IP-based blocking gives only light security.

In a GDIIS, system specific properties can be used to drop unauthorized connection attempts. The message flows are often deterministic: size, frequencies and sequences of the messages are known beforehand. Furthermore, if the communication is not urgent in timing, the connection can be restricted to certain time windows. It might be also possible to give only to certain groups direct access to the remote machine, while others must communicate through a less valuable web server. (Naedele,)

User authentication. The management of usernames and passwords can be challenging in a GDIIS, and hence instead of user-based authentication, role-based authentication should often be used. Without role-based authentication, always when a new user is added to the system, every remote devices' password files must be updated. In the role-based authentication, the username and password must be sent to the new user.

Action authentication (authorization). Action authentication is controlling which actions and action sequences a user is allowed to do. This is often tightly related with services, software and the underlying operating system. Therefore, action authentication is often handled by those. Application level firewalls are counted as action authentication, too. They can be used to manage and monitor application level communication. (Naedele,)

Intrusion detection

Virus detection, network traffic monitoring and authentication failure detection are all needed parts of a GDIIS's intrusion detection. In addition to those, system files and file system integrity as a whole can be checked against unauthorized change (Naedele,).

Encryption

The NIST (National Institute of Standards and Technology) recommends that 128-bit protection should be used to achieve relatively long lasting security

(up to the year 2036) (Vanstone, 2003). In symmetric encryption this means moving from popular 3DES (Data Encryption Standard, regarded to provide 112 bit security (Vanstone, 2003)) to AES (Advanced Encryption Standard, where security level increases as the key size increases (Anderson, 2001, p. 93)).

128-bit security is more challenging in traditional asymmetric algorithms (e.g. Rivest, Adi Shamir (RSA) and Digital Signature Algorithm (DSA)). According to (Vanstone, 2003), to provide 128-bit security, those algorithms need 3072 bit keys, and to provide 256-bit security a 15360 bit key is needed. However, those keys are too heavy for the remote devices. The problem gets even further complicated as the greater key size leads to even greater increase in computational cost (Vanstone, 2003).

In contrast, with elliptic encryption, the key size is only double to the security level. Therefore, the elliptic curve-algorithms have definite advantages especially in those devices which are limited in computational power, storage space, bandwidth or power consumption. (Vanstone, 2003)

A deeper study on PKI, Elliptic Curve Cryptography, and Digital Signatures is found in (Caelli et al., 1999).

4. THREE SECURITY ARCHITECTURES FOR THE GDIIS

In the previous chapters we discussed the assets and threats of a GDIIS as well as security mechanisms to protected the assets. In this chapter, we examine three security architectures for the GDIIS. In the first architecture, all connections are centralized in order to ease management and use of security mechanisms. In the second architecture, the GDIIS is divided into zones, and instead of relying on only one solution, also security is distributed. In the third architecture, the centralized and the layered (zone) architectures are integrated.

Centralized connections

In the centralized connection architecture, all connections from the human interface to the remote devices go through a server, a stepping board. Behind the stepping board there can be many remote devices or device groups which are geographically apart from each other. Hence, the centralized connection architecture consists of three parts *human interface*, *remote device* and *stepping board*. (figure 2)

Human interface. We can place the human interface authentication and authorization functionalities in the stepping board. In this way, all user passwords can be managed at the same place (at the stepping board). The stepping board makes it also possible to use sophisticated and heavier security mechanisms

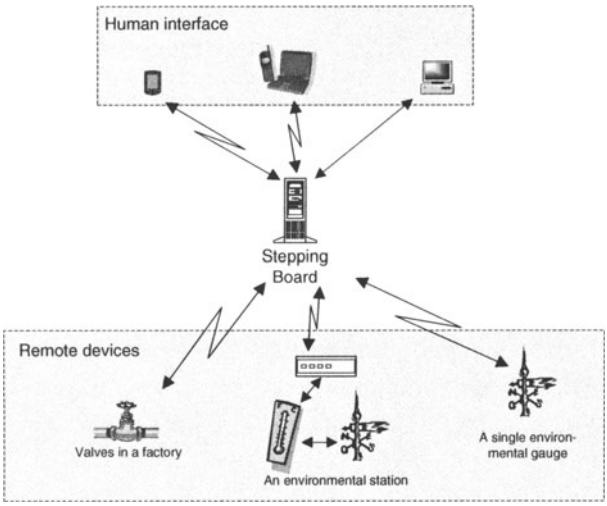

Figure 2. Centralized connection architecture

between the human interface and the stepping board, such as smart cards or AES encryption. Installing the counterpart of the smart card-system to the every remote device is often unpractical because of economic and management reasons, whereas AES is often too heavy for the remote devices.

The human interface devices can be anything from a mobile phone to a desktop computer. The capabilities of these (e.g. display) demand different servicing. In addition to content specific servicing, also security mechanisms may differ. For a mobile phone, we could use lighter encryption and allow the user to access only monitoring services. For a desktop computer, we could use stronger encryption and also enable control services. Because of needed resources for such a service, we cannot really place those in the remote device.

Remote device. We can utilize proprietary security mechanisms between the stepping board and remote devices, which means, that stronger security is achieved by obscurity. However, proprietary security mechanisms mean here more taking advantage of known and deterministic communication than building a really new mechanism. The use of proprietary solutions directly between the human interface and the remote devices is harder, because of the heterogeneity and changeability of the human interface devices. Further, we can also take the advantage of the fact, that the stepping board is trustworthy from the remote device's viewpoint. (Even if the human interface was properly authenticated, it could always try to do unauthorized things whereas the stepping board most likely would not).

Stepping board. Because all communication goes through one computer, we can utilize centralized IDS in the stepping board. The network traffic can be scanned for viruses and other anomalies. We can place a sophisticated firewall in the stepping board to protect remote devices from DOS attacks and port scanning. Also, logs and other information from all remote devices can be integrated at the stepping board to get the general picture of the whole system.

When the stepping board can act as a proxy server between the remote devices and the human interface, we can prohibit direct access to the remote devices. Sometimes it might even be possible to make the connection between the remote devices and the stepping board only one-way. A remote device only pushes its data to the stepping board. If we can ensure the one-way communication (e.g. by hardware), monitoring of very critical remote devices can be enabled, too.

Generally, the security mechanisms in the stepping board can be much more sophisticated and heavier and hence give stronger security. However, this does not mean that we can leave out all security mechanisms in the remote devices. The remote devices must still have their security mechanisms, but those can be lighter.

Layered architecture

A common architecture to protect the assets is to divide the system into trusted and untrusted zones and place a wall between them (e.g. intranet from the Internet). Traditionally, the separation of the zones is achieved by a firewall (e.g. a stepping board) which monitors the connections to and from the trusted zone. However, this type of *hard perimeter* solution has severe problems.

Firstly, the wall must have doors to and from the trusted zone. Because of complex programs, the wall has always weak points and unknown doors. Those might (will) be used by an attacker. (Naedele,)

Secondly, building the solution on only one principle is dangerous: if that principle fails, the whole solution fails. Once the attacker gets inside, the system is without security and everything is already lost. Hence, the wall should be infinitely strong to last for an infinitely long time. (Naedele,)

Lastly, technology is developing all the time, which gives attackers better penetration tools. (Naedele,)

Accordingly, instead of dividing the GDIIS into two zones, we should divide it into several security zones. This type of *defense-in-depth* architecture is used e.g. in automation systems of nuclear power plants. In this way, the core services and assets can be placed into the innermost zone, which then has several security zones around it. Basically, the system is in safe as far as the formula $P > D + R$ holds, where P is the time which is needed to break the

security, D is the delay until the attack is detected and R is the time until a reaction against the attack has been completed (Schwartau, 2001).

When there are many zones, the administrator has better capabilities to fight the attacker. The compromisation of the first zone can raise alarms. When actions are taken early enough, the attack can be stopped.

To utilize the *defence-in-depth* architecture in the GDIIS, we can divide it into four zones: inter-zone, intra-zone, remote device -zone and application-zone. The inter-zone is an open and insecure global network. The intra-zone is open and insecure as well, but is usually national and managed by a trusted party (e.g. a ISP's GSM network). The remote devices lie in the remote device -zone. The information and the core services of the GDIIS are in the application-zone.

Inter-zone. In the GDIIS, the inter-zone is usually the Internet and can be described with the adjectives untrustworthy, globally reachable and heterogeneous. On the other hand, very light security, openness and globality are some of the Internet's strengths: it keeps the core protocols and routing simple and effective and connects almost everyone to almost anywhere. However, this unsecureness is a problem for the GDIIS.

To make the Internet more secure does not automatically mean heavy encryption or disposal of openness and globality. We could place some light and sophisticated intrusion detection systems in the backbone network. This could be, for example, the detection of the newest viruses and simple DOS detection systems. However, our goal is not to make the Internet 100% secure, but to detect the most common attacks by means of simple and light security mechanisms.

Placing security mechanisms in the backbone network of the Internet would cost. The costs could be charged from all Internet users thorough ISPs, and because there are millions of Internet users, the cost per user would not be high, especially when compared to the benefits. An other possibility is to build secure tunnels over the Internet and sell those tunnels for additional cost. From industry's viewpoint, either one would be acceptable as long as the costs are kept reasonable.

Intra-zone. The intra-zone can be defined as a network hosted by one ISP. Usually, the intra-zone is an open network, such as the Internet, but we can often restrict access to certain user groups and use more sophisticated security mechanisms. In the GDIIS, the intra-zone is often a GSM/GPRS network, but satellite networks are also used. Because of the popularity of GPRS network, we study the intra-zone from GPRS network's viewpoint.

Brief architecture of GPRS-network is found in figure 3 (Chang, 2002). A more detailed description and special security threats in GSM network can be found in (Rautpalo, 2000) and (Chang, 2002).

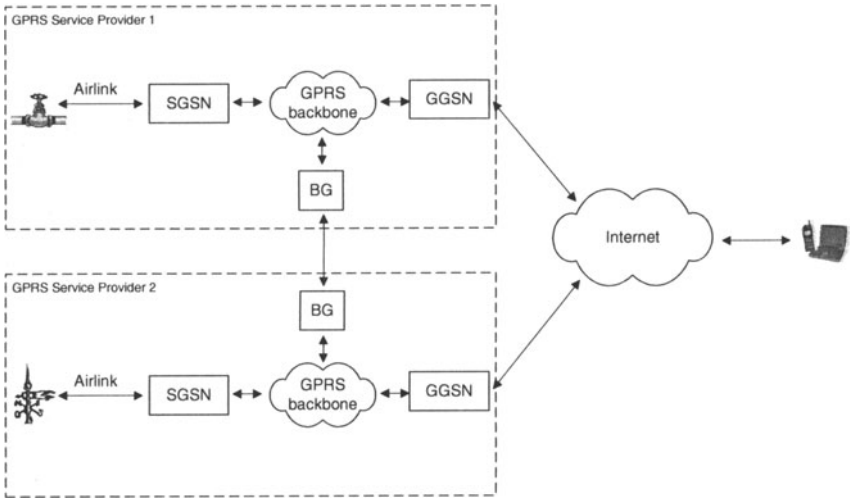

Figure 3. Arcihtecture of a GPRS network

SGSN Servicing GPRS Support Node - A gateway between a GPRS and a GSM network

BG Border Gateway - A gateway between two GPRS networks

GGSN Gateway GPRS Support Node - A gateway between a GPRS-net -work and the Internet

In chapter 3, we divided authentication into three categories: connection, user and action authentication. The last of these is tightly related with the service, hence it is not appropriate in the intra-zone.

We can place some connection authentication mechanisms in GGSN (Gateway GPRS Support Node). When it is known beforehand from which IP-address(es) certain network blocks or individual devices are accessed, we can drop unauthorized connection attempts. However, if there are public services accessible from any IP address, it is hard to place IP-based blocking in GGSN. In Border Gateway (BG), IP-based blocking is easier, because the address spaces of both networks is known.

For user authentication, GPRS utilizes a A3 GSM authentication algorithm. A unique subscriber authentication key is stored in a physical Subscriber Identity Module (SIM). Hence, to gain access to the GPRS network, a user must have an authorized SIM card. Compromisation of a SIM card is hard. Another

means of gaining access to the GPRS network is to steal an authorized card. Thus, as long as the SIM card cannot be stolen, authentication in GPRS is sufficient for most purposes. (Chang, 2002) In some cases, we could utilize GSM A3 user authentication as such in the GDIIS.

Dealing effectively with viruses is today's challenge and industry welcomes all well-working IDS services. It has been seen that traditional anti-virus software in a mail gateway decreases virus problems dramatically. We could adopt the scanning technique in the other traffic as well.

In addition to virus detection, we could prevent port scanning of the remote devices in the GGSN. Today's port scanning methods may last months and therefore need logs over long time periods. Further on, DOS is one of the biggest threats against availability. Even heavy servers are in great danger, all the more the lightweight remote devices.

After the authentication procedure, the connection from the remote device to SGSN (Servicing GPRS Support Node) is encrypted with the GPRS-A5 algorithm (Chang, 2002). According to (ETSI,), the compromisation of that is hard even with massive computing resources. An other advantage is, that this encryption algorithm suits well the remote devices, because it is designed for resource limited mobile phones. However, the utilization of this algorithms for other purposes, (e.g. end-to-end encryption) is not possible, because the algorithm is not public.

In the GPRS backbone from SGNS to GGSN (or BG) special GPRS Tunneling Protocol (GTP) is used. GTP has no security mechanisms. However, the GTP is not totally insecure, because the communication occurs over a private network (Chang, 2002). Additionally, the ISP can encrypt the information in the backbone. Hence, the encryption in the GPRS network is enough for most purposes as long as the ISP's own personnel cannot access the information.

Earlier described NAT is often a default in today's GPRS networks, because of the lack of IPv4 addresses (Rautpalo, 2000). NAT protects the remote devices against direct access from the Internet, but does not protect against unauthorized access from inside the GPRS network. However, the problem of NAT is that it prevents all direct access to the remote devices (also the authorized one). Therefore, NAT is also a problem, because the connection is often initiated from the Internet. However, we can use other techniques (e.g. SMS) to tell the remote device to initiate the connection.

NAT makes it also difficult to use some end-to-end encryption mechanisms. For example, IPSec packets will be dropped, because the integrity check fails (address changes in the header). One solution for that is to encapsulate IPsec packets in UDP, which can be used without the integrity check. The drawback of UDP is that the additional encapsulation and possible key management slows down the setup and the connection itself. (Rautpalo, 2000).

Remote device-zone. The remote device -zone is the last lock before the core services. The surrounding zones can prevent some attacks from happening, but not all. Actually, we must still be prepared to face any attack in the remote device -zone. The task of ensuring security becomes nearly impossible when talking about e.g. a single pump which should be directly controlled from the Internet. Luckily, the situation is seldom like that.

In all cases, we must authenticate connections and users and need some encryption mechanisms. Also, we should place some level of malicious code detection in the remote device -zone. We can utilize the time window technique such that the network interface of the remote device is turned off, when the service are not needed. In general, because the remote device is near the core services, we can and should utilize the known properties of the communication in the security mechanisms.

We should also protect the security mechanisms themselves. We can easily disable unnecessary services. This saves us from many surprises. We can also download the core configuration parameters regularly from a secure server. Furthermore, we can utilize component based solutions (e.g. Java Embedded Server or Open Service Gateway Initiative), where all services are separate modules on top of operating system. This makes it possible to add, remove and update the services easily and safely.

To conclude, it might be impossible to reach the desired security level in the remote device -zone. Then we should firstly utilize every security mechanism, which the limited resources permit. Secondly the level of services should be reconciled with the achieved security level.

Application-zone. The application-zone is the most critical part of the system. However, there is not much we can do at this level. Here, the emphasis is particular on action authentication: who is allowed to perform which actions. This can be an inherent part of the underlying operating system, can be handled by the application, or both. In any case, we need action authentication, because by breaking the remote device -zone the attacker would automatically have full control of the device.

In the application -zone, service specific properties are known and can be used to increase security. We can utilize those e.g. to drop unauthorized connection attempts or to detect anomalies in communication. Also, if possible, we do not send the messages in self-explaining format, such as xml. To conclude, appropriate mechanisms in the application-zone are rare and utilize often service specific properties.

Integration of centralized and layered architecture

Some problems exist in the both above proposed architectures. The centralized architecture is like a hard perimeter solution. In the layered architecture,

the biggest problem is that the whole path from the remote device to the human interface is still assumed to be insecure. Here, the remote device often becomes the bottleneck and hence reduces the end-to-end security mechanisms to be light.

Most presumably the biggest threats come from the inter-zone. Hence, if we can place the stepping board in the border of the inter- and intra-zone, we can use lighter security mechanisms in the inter-zone and heavier ones in the intra-zone. That is, we utilize some light and proprietary security mechanisms between the remote device and the stepping board and the heavier ones between the stepping board and the human interface. Therefore, the challenges in the inter-zone (human interface - stepping board) are solvable by standard security mechanisms.

In the intra-zone, perhaps the best location for the stepping board is at the border of the GPRS network and the Internet (figure 4). In the stepping board, we can utilize security mechanisms of GGSN in more suitable way for the GDIIS. Encryption of GPRS can be enough, but some level of connection, user and action authentication is still needed, because the connection attempts can always come from the inside of the intra-zone. Because only the stepping board is allowed to connect to the remote device, the stepping board-remote device -authentication can be simple and strong at the same time. This fact also simplifies also the connection authentication (all other attempts can be dropped except the stepping board).

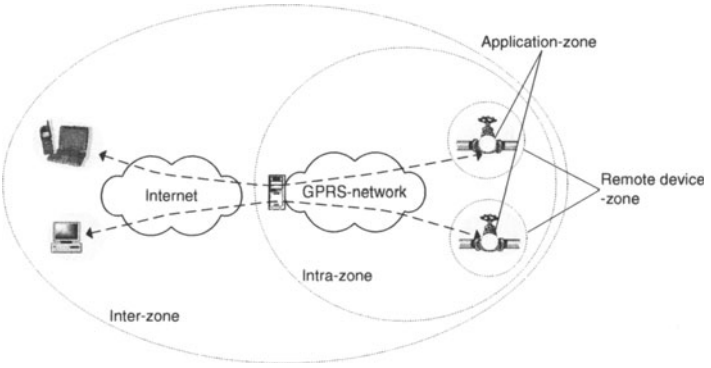

Figure 4. Integrated connection architecture

When the stepping board is in an ISP's network, the ISP can provide the stepping board as part of the service. Also, the ISP can join and analyze the information from both the stepping board and GGSN. But, even if it were possible to integrate the stepping board and GGSN, it should not be done, because the GGSN serves all of the ISP's customers. Therefore, the integration of the stepping board and the GGSN would reduce the benefits gained.

5. DISCUSSION

We have now examined three different architectures. The biggest advantages of *the centralized architecture* is its centralized security and the positive consequences of it. Management of passwords becomes easier and centralized IDS is possible. Between the stepping board and a remote device, we can utilize lightweight and proprietary security mechanisms, whereas heavier ones can be placed between a human interface and the stepping board. Also, human interface specific servicing is possible (e.g. controlling services only for PC). Further, with the stepping board, we might be able to ensure one-wayness of the connection, which makes it possible to monitor also very critical remote devices.

However, the stepping board is a hard perimeter solution, i.e. it rests only on one security principle. Also some of the advantages might be only theoretical. Firstly, the network between the stepping board and the remote devices can be so insecure that everything has to be built twice. Secondly, it can be impossible to place the stepping board close enough to the remote devices to really protect them, because we should have enough remote devices for a stepping board to make the architecture economically viable.

With *the layered architecture*, we can reduce the problems of hard perimeter solution. In the layered architecture, the security mechanisms in the limited remote devices can be lighter. A layered architecture makes it also possible to have different parties to manage different zones. From the industrial viewpoint, it would be nice to be able to buy all other zones except the remote device and the application -zone. Then, industry could concentrate on its core competence without having to study new security mechanisms.

A drawback of the layered architecture is that when we examine it in detail, it might not seem to protect the remote devices more effectively than the centralized architecture. We must still be prepared to face actually any threat in the remote device -zone, because the intra-zone is also a public network. Therefore, the security of the intra-zone plays a big role. With only little security in the intra-zone, we might be forced to reduce the level of services of the GDIIS (e.g. from control to monitoring).

We can partially solve the problems of the hard perimeter solution (centralized architecture) and the limited end-to-end security mechanisms (layered architecture) by *integration of the two architectures*. The layered architecture automatically removes the problem of hard perimeter solution. Further, with the stepping board we can place stronger security mechanisms in the inter-zone without loading the remote devices excessively.

That said, even integration leaves some problems. The topology of GPRS networks is not necessarily such that the stepping board really protects the remote devices. Also, the remote devices cannot be distributed globally, be-

cause the GPRS networks are mostly national. Furthermore, the stepping board means (almost) necessarily longer paths between a remote devices and a human interface.

If the ISP can (in addition to the intra-zone), manage by itself or through a third party, the stepping board and the inter-zone, the industry could outsource the management of the inter- and intra-zones to the ISP. The drawback of this approach is that the ISP must be trustworthy. However, we must not trust the ISP blindly. The ISP should strictly restrict of who its personnel and how is allowed to administrate the stepping board. All changes should be accountable and non-repudiable and the ISP's system must be audited by a trusted third party.

Table 1 shows, which security mechanism should be applied in which zone. From the industry viewpoint, the intra-zone should include all computational power and deep security knowledge demanding security mechanisms, which are complemented by the mechanisms in the remote device-zone.

Security mechanism	**Inter-zone**	**Intra-zone** (stepping board)	**Remote device -zone**	**Application -zone**
Connection authentication		X	X	
User authentication		X	X	
Action authentication			(X)	X
Intrusion detection	(X)	X	X	(X)
Encryption		X	X	(X)
Security mechanism protection		X	X	

Table 1. Appropriate security mechanisms in each zone.

6. CONCLUSIONS

Development in Information Technology has made it possible to move from centralized systems to distributed ones. Today's industrial devices, e.g. valves, pumps and rainfall gauges, have remote control and monitoring capabilities. However, one of the biggest hindrances for further development is security. There are assets to be protected, i.e. information, services and remote devices themselves, as well as software. Threats coming from the Internet are real and challenging. Because remote devices have limited resources, standard security mechanisms cannot be applied as such.

When we compare the communication of a GDIIS and the Internet, we find that the properties of the GDIIS's communication are often more deterministic.

Among others, we might be able to predict the content, frequency and paths of the messages. We can and should utilize those to increase the security, especially in the near of the core services (the remote device-zone and the application-zone).

We studied three possible architectures for the GDIIS, from which the integration of the centralized and the layered architecture seems to have the most advantages. Security mechanisms at the border of the inter zone (Internet) and intra zone (GPRS) should be such that the inner zone's mechanisms can be lighter to save remote devices' limited resources. Hence, the integrated architecture applies the general principles of defense in depth. Even if the outer zones have very strong and sophisticated protection, those will eventually fail and security in the inner zones is needed. The inner-zone security protects also against threats coming from inside the system.

To conclude, the security in a globally distributed industrial information system is a challenge. But, instead of trying to place security in one place, we should distribute the security also. We should see the distributed system as a whole and have security mechanisms as a integrated parts of different zones, not as add-on features.

Acknowledgments

We would like to thank Teemu Tommila for his patient guidance and Elina Valtonen for sharing her (unpublished) evaluation of earlier work in this area.

References

Anderson, Ross (2001). *Security Engineering*. John Wiley & Sons, USA.

Caelli, William J., Dawson, Edward P., and Rea, Scott A. (1999). PKI, elliptic curve cryptography, and digital signatures. *Computers & Security*, 18:47–66.

Chang, Dung (2002). Security along the path through gprs towards 3g mobile telephone network data services, version 1.3. SANS Institute 2002, As part of the Information Security Reading Room, referenced 27.9.2003.

ETSI, European Telecommunications Standards Institute. GSM calls even more secure - thanks to new A5/3 algorithm. referenced 17.10.2003.

Naedele, Martin. IT security for safety-critical automation systems. ABB.

Rautpalo, Jussi (2000). GPRS security - secure remote connections over GPRS. Technical University of Helsinki, Department of Computer Science, referenced 27.9.2003.

Schwartau, Winn (2001). Network security it's about time: An offer for a metric. *Network Security*, 2001:11–13.

Vaisala. National lightning detection network. referenced 24.10.2003.

Vanstone, S.A. (2003). Next generation security for wireless: elliptic curve cryptography. *Computers & Security*, 22:412–415.