Security Analysis of Open Building Automation Systems

Wolfgang Granzer and Wolfgang Kastner*

Vienna University of Technology,
Institute of Computer Aided Automation, Automation Systems Group
Treitlstr. 1–3, 1040 Vienna, Austria
{w,k}@auto.tuwien.ac.at
https://www.auto.tuwien.ac.at/

Abstract. With the integration of security-critical services into Building Automation Systems (BAS), the demands on the underlying network technologies increase rapidly. Relying on physically isolated networks and on "Security by Obscurity", as it is still common today, is by no means an adequate solution. To be reliable and robust against malicious manipulations, the used communication services must support advanced security mechanisms that counteract potential security threats. This paper identifies important security requirements and challenges within the building automation domain. Based on this analysis, state-of-the-art technologies are carefully examined. Finally, an outlook on advanced security concepts is given.

Key words: Building Automation, Embedded Networks, Security

1 Introduction

Building Automation Systems (BAS) aim at improving control and management of mechanical and electrical systems in buildings – more generally, interaction among all kinds of devices typically found there. The core application area is the automatic control of traditional building services like lighting/shading as well as Heating, Ventilation, and Air Conditioning (HVAC). Services from the security domain (e.g., intrusion alarm systems, access control) are often provided by separated, application-specific subsystems.

Today, a trend towards the integration of these separated subsystems into the core BAS can be observed. The advantages of such a resulting "all-in-one" BAS are manifold. First, the application area can be extended since services from the security domain can also be served by such an all-in-one system. Second, traditional services like HVAC and lighting/shading are also improved since a comprehensive security concept will also protect the BAS against among others vandalism acts.

* This work was funded by FWF (Österreichischer Fonds zur Förderung der Wissenschaftlichen Forschung; Austrian Science Foundation) under the project P19673.

To be able to fulfill the requirements of such a secure all-in-one BAS, the underlying technologies must be reliable and robust against malicious manipulations. However, available BAS installations rely on physical isolation and "Security by Obscurity". This is obviously unacceptable within modern BAS since preventing physical access to the network by isolation is not always possible (e.g., WLANs, public buildings). Moreover, "Security by Obscurity" is a technique that (if at all) provides only temporary protection.

This paper provides a comprehensive analysis of the integrated security concepts of available BAS solutions. First, important security requirements as well as domain-specific challenges are identified. Based on these requirements, existing standards within the BAS domain are analyzed. This analysis is focused on the most important open BAS standards (i.e, BACnet, LonWorks, KNX, and ZigBee). The paper is concluded with a summary about the opportunities and drawbacks of today's BAS technologies regarding their suitability within security-critical environments.

2 Security requirements and domain-specific challenges

To be able to serve as a BAS solution for security-critical environments, the used network technologies must fulfill different security requirements. Based on [1–3], the following *Functional Requirements (FR)* for secure BAS are identified. First, the communication entities that want to securely exchange data (e.g., sensors, actuators, controllers, management devices) must prove their identities i.e., it must be verified whether the entities are what they claim to be (*entity authentication FR1*[1]). Then, it must be verified if the entities have the necessary access rights to participate in the communication (*authorization FR2).* Afterwards, the data exchanged between authenticated entities must be protected in a secure manner. This is done by establishing a so called *secured channel.* A secured channel uses non-cryptographic (e.g., physical or organizational measures) and/or cryptographic techniques to protect data against security attacks while they are transmitted over a network. Depending on the requirements of the application, a secured channel guarantees the following security objectives:

- *Data integrity (FR3)* proves that the data was not modified.
- *Data origin authentication (FR4)* is a stronger form of data integrity where a receiver can also verify the data origin i.e., the data source.
- *Data freshness (FR5)* guarantees that the transmitted data is recent and valid at the time of transmission. Replaying of previously sent data can be detected by the entities.
- *Data confidentiality (FR6)* ensures that only authorized entities have access to confidential information. A typical example of confidential information would be a PIN code that is entered by a user at a security door.
- *Data availability (FR7)* guarantees that the communication is not disturbed and that the authorized entities have access to the data.

[1] This numbering style is used throughout the paper to uniquely identify the different requirements and challenges.

Besides these functional requirements, various *Domain-Specific Challenges (DC)* that reflect the characteristics of the environment exist. They are the main reasons why it is not possible to directly use security concepts from other domains. For example, mapping Information Technology (IT) security mechanisms to the BAS domain is not possible in a native way since they are tailored to the use in the IT world. This is also true for closely related domains like industrial automation. The domain-specific characteristics of BAS lead to the following challenges. BAS typically consist of embedded networks where *low-power embedded devices (DC1)* are used. Due to reasons of cost and space efficiency, these devices are equipped with limited system resources. This concerns the amount of available memory, processing power but also the power supply (e.g., bus-, battery, or self-powered devices). However, since security mechanisms are computationally intensive (especially cryptographic algorithms), their use must not exceed the available resources. Therefore, it is essential to find a good balance between a required level of security and available resources ("*good enough security*"). For example, if the non-disclosure of the transmitted data is not strictly necessary, data confidentiality is unnecessary.

An important difference between BAS and communication systems within other domains is the required support for different *communication models (DC2)*. While in other domains the client/server model is predominantly used, group communication patterns based on multicast or broadcast are well-established in the BAS domain. This also concerns the amount of devices used within a network. BAS usually consist of hundreds or even thousands of devices. Thus, *scalability (DC3)* of the integrated security mechanisms and security management services (e.g., distribution of secret keys) is of major concern.

IT security mechanisms are geared towards different requirements regarding the used network technology. While in the IT world IP based network protocols are dominant, *non IP field networks (DC4)* are mainly used at the field level within the BAS domain. The main reasons for the use of such networks are robustness, flexibility, and cost efficiency.

Finally, the required *Quality-of-Service (QoS) (DC5)* parameters of BAS field networks differ from the IT/office world, too. In the IT/office domain, the data volume to be transferred is commonly high (in the order of mega- or gigabytes) with usually no real-time requirements. Control data typically transmitted in BAS has a small volume (in the order of some bytes) with perhaps soft real-time requirements (e.g., the reaction time of a lighting system). Additionally, QoS properties like *reliability* and *ordering* of messages have to be considered. While these QoS properties are normally of less concern in the IT/office world, they may be an important issue in the BAS domain.

3 Security in home and building automation standards

Today, many different BAS protocol standards exist. The most important open ones that can be considered as reasonable solutions for all-in-one systems are BACnet [4, 5], LonWorks [6, 7], KNX [8, 9], and ZigBee [10].

3.1 BACnet

BACnet offers several services which pretend to provide support for data confidentiality, data origin authentication (and thus data integrity), and data freshness as well as entity authentication [4, 11]. Authorization is provided on a per-device basis. The security mechanisms are based on Data Encryption Standard (DES) and a trusted key server which is responsible for managing session keys. These session keys are used to secure the transmitted data between two devices. To establish a secure connection to the key server, each device must own an initial secret key.

Due to several security flaws [11–13], this security concept was completely replaced by a new one that is defined in BACnet Addendum g [14]. At the time of writing, BACnet Addendum g has finished the 4th public review process and is now waiting for final publication. BACnet Addendum g specifies security services that are designed to be applicable to all BACnet media and device types. To protect the transmitted data, symmetric cryptographic algorithms are used exclusively. The required shared secret keys have to be distributed in advance or they have to be retrieved from a so called *key server* during runtime. In BACnet Addendum g, six different key types are distinguished. `General-Network-Access` keys are shared between all members of a network. `User-Authenticated` keys are used for requests where the user identity can be assumed to be authenticated properly. The user authentication has to be performed by an external mechanism (e.g., via a user interface). Alternatively, if a device does not have a user interface, the user identity can be configured directly at the device. `Application-Specific` keys are dedicated to a dedicated application domain (e.g., HVAC or access control). These keys are only distributed to a subset of devices that require a higher level of security. `Installation` keys are temporally used for management purposes. `Distribution` keys have the aim to secure the retrieval of other keys from the online key server. Finally, `Device-Master` keys are only used to receive `Distribution` keys. Since they act as initial secrets, their distribution must be done within a physically secured environment.

BACnet Addendum g specifies eight secure communication services that are incorporated into the network layer of BACnet. The `Security-Payload` service is used to securely transmit data messages. To respond to them, the `Security-Response` service is available which indicates either the successful retrieval of a secured message or an error condition. The `Challenge-Request` service is used to verify the identity of a device. The device that is challenged has to answer with a `Security-Response` message that contains the result of the challenge. To request the distribution of the secret keys from the key server, the `Request-Key-Update` service is available. Upon retrieval of a `Request-Key-Update`, the key server responds with an `Update-Key-Set` or with an `Update-Distribution-Key` message which contains the requested key set. These two services can also be used by the key server to force key changes. Finally, the `Request-Master-Key` and `Set-Master-Key` are used to change the `Device-Master` key. However, since

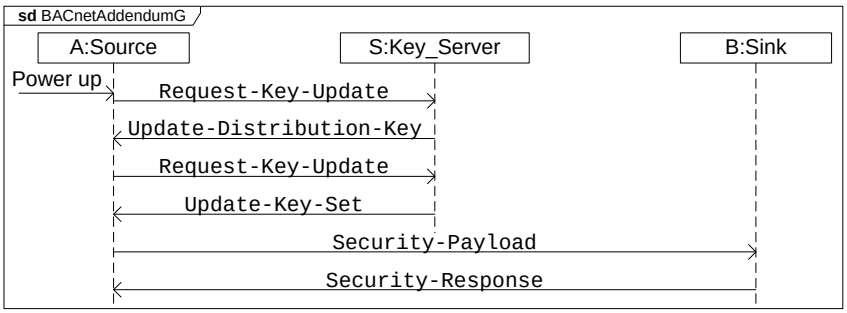


Fig. 1. Security services in BACnet Addendum g

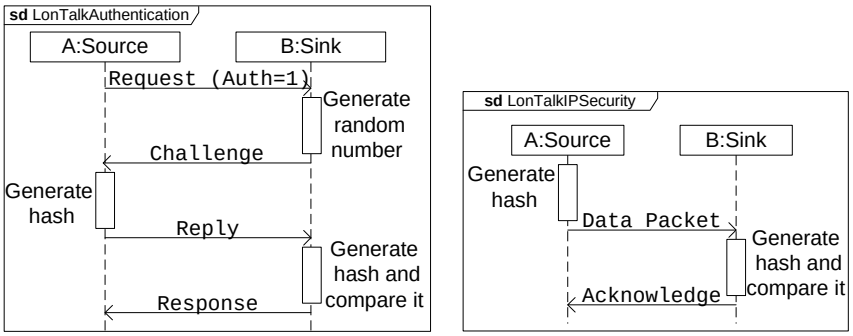
these two services are not secured at all, their use has to be limited to physically secured environments.

Fig. 1 shows an example how these security services can be used. After having powered up, device A requests a `Distribution` key from the key server S by sending a `Request-Key-Update` message (secured with its `Device-Master` key). The key server validates the request and transmits a newly created `Distribution` key to A. Afterwards, A sends another `Request-Key-Update` message to retrieve the current keys. This request is secured using the `Distribution` key retrieved before. After having received the key set from the key server, A is now able to securely communicate with device B using the appropriate key. Note that it is assumed that device B is also in possession of the used key (e.g., `General-Network-Access` or `Application-Specific` key).

Network messages are classified into *plain*, *signed*, and *encrypted* messages. While plain messages are not secured at all, signed messages provide data integrity and freshness. To guarantee data integrity, Keyed-Hash Message Authentication Code (HMAC) in combination with Message Digest 5 (MD5) or Secure Hash Algorithm (SHA) is used. Data freshness is achieved by using a timestamp (32 bit standard UNIX timestamp) in combination with a 32 bit message ID. Encrypted messages are additionally encrypted using Advanced Encryption Standard (AES) in Cipher Block Chaining (CBC) mode. Entity authentication is implicitly guaranteed due to the used symmetric algorithms and due to the use of so called device instance numbers. Device instance numbers uniquely identify secure BACnet devices and are assigned to the devices independently.

3.2 LonWorks

The communication protocol of LonWorks (called LonTalk) provides a rudimentary security concept based on a four step challenge-response protocol. During this protocol, the identity of the sender is verified. Furthermore, it pretends to guarantee data integrity and freshness. Fig. 2(a) shows the different steps: a sender which desires to secure a request sets the so called authentication bit of



(a) Four step challenge-response protocol (b) IP tunneling security mechanism

Fig. 2. LonTalk security mechanisms

the corresponding message. All receivers have to reply with a 64 bit random number. The sender receives these random numbers and individually calculates a 64 bit hash value over the content of the message and the random number using a shared secret key. These hash values are sent back to the receivers where the same calculation is performed and compared with the previously received value.

In addition to the basic challenge-response protocol, the IP tunneling scheme of LonTalk defines its own security mechanism (cf. Fig. 2(b)). It uses MD5 together with a shared secret to calculate a hash value. This hash value is sent together with the message to the intended receiver(s). After having received a secured message, the receiver calculates its own hash value using the same shared secret and compares it with the received one. If both values are equal, the message is accepted – otherwise it is discarded. Note that the four step challenge-response mechanism mentioned above is not used here.

3.3 KNX

KNX only provides a basic access protection scheme based on clear text passwords (cf. Fig. 3). Up to 255 different access levels can be defined, each of them is associated with a different (otherwise unspecified) set of privileges. Access level 0 has the highest privilege and access level 255 is the lowest one. For each of these access levels, a 4 byte password can be specified. This scheme is only available for engineering communication. Control data exchange remains insecure.

To be able to use IP networks for KNX installations, KNXnet/IP has been introduced. In the corresponding specification [8], some rudimentary security guidelines are additionally presented. These guidelines are based on network isolation (e.g., use of firewalls or KNXnet/IP only Intranets) and on "Security by Obscurity" (e.g., use of non-standard IP addresses, rely on the missing expertise of an attacker). Since preventing physical access to the network by isolation is not always possible (e.g., WLAN) and "Security by Obscurity" is a technique

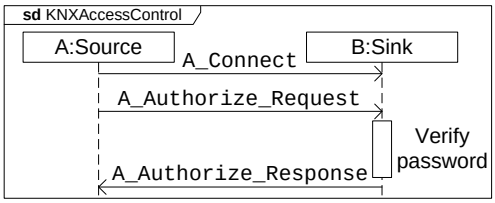


Fig. 3. Access control mechanism of KNX

that (if at all) provides only temporary protection, these security guidelines do not provide an effective protection.

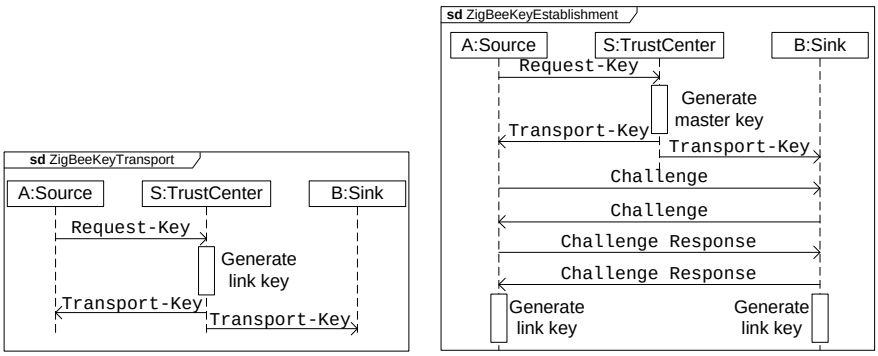
3.4 ZigBee

ZigBee (version 2007) is the most well-known protocol that builds upon IEEE 802.15.4. ZigBee uses the data link layer of IEEE 802.15.4 (version 2003) and enhances the available features by specifying an application layer and a network layer. Additionally, new services that provide the opportunity for multi-hop routing and advanced security services have been added. Although, while ZigBee uses the transmission services of the data link layer of IEEE 802.15.4, it defines its own security architecture that is independent from IEEE 802.15.4. Thus, the security services provided by IEEE 802.15.4 are entirely not used.

The security concept of ZigBee is exclusively based on symmetric cryptographic schemes. In particular, AES and a variant of Counter with CBC-MAC (CCM*) are used. Entity authentication as well as data origin authentication, freshness, and confidentiality are provided at the network and/or application layer. Additionally, ZigBee provides services for management and distribution of the required shared secret keys. Depending on their use, ZigBee distinguishes three different key types. *Link keys* are shared between two devices. They are used to secure communication between them. *Network keys* provide security across the whole network segment. Finally, so called *master keys* are optionally available. Master keys are only required during the establishment of link keys.

Beside the possibility to manually install shared secret keys in advance, it is possible to retrieve secret keys during runtime. This runtime distribution of shared secret keys is handled by a single entity called *Trust Center*. To exchange secret keys, three different distribution methods are available in ZigBee:

- *Pre-installation*: Here, the keys are uploaded to the devices before runtime using, for instance, a proprietary management tool. The exact method how pre-installation is performed is not defined by the ZigBee specification.
- *Key-transport*: Using key-transport, the trust center sends the keys directly to the devices using a dedicated communication service. Key-transport is used to distribute the actual network key during the device joining process and to distribute link keys during runtime. Fig. 4(a) shows an example how



(a) Key transport (b) Key establishment

Fig. 4. ZigBee security mechanisms

key-transport can be performed to distribute a link key. To retrieve a link key, the initiating device sends a `Request-Key` message to the trust center. The trust center generates a new link key and distributes it to both devices using a `Transport-Key` message. The message is secured with the trust center link key that is shared between the trust center and the corresponding devices.

– *Key-establishment*: Key-establishment is only available for link keys. In contrast to key-transport, both devices are involved in the key generation process. The key-establishment is performed using the so called Symmetric-Key Key Exchange (SKKE) protocol (Fig. 4(b)). To start the key-establishment process, the initiating device sends a `Request-Key` message to the trust center. The trust center generates a master key and distributes it to both devices using the `Transport-Key` service. Afterwards, the devices start the SKKE protocol. First, each device generates a random challenge that is sent to the other device. Using this challenge and the previously retrieved master key, each device calculates a challenge response which is sent to the other device. After having retrieved the challenge response, both devices verify it. If it is valid, a link key is calculated out of both challenges which can later be used to secure the communication between the two devices.

To be able to securely retrieve network, master, or link keys from a trust center, the requesting device must share a link or master key with the trust center. These initial trust center keys can either be pre-installed or distributed using unsecured key-transport messages. However, in the latter case, the exchange has to be done in a physically secured environment.

4 Evaluation

To evaluate available BAS standards regarding their suitability within security-critical environments, their integrated security concepts were analyzed with respect to the requirements and challenges identified in Section 2. Fig. 5 sum-

	BAS				IT mechanisms	
	BACnet	LonTalk	KNX	ZigBee	IPsec	TLS
Entity authentication (FR1)	+	-	-	+	+	+
Authorization (FR2)	∼	-	∼	∼	+	+
Data integrity (FR3)	+	∼	-	+	+	+
Data origin authentication (FR4)	∼	-	-	+	+	+
Data freshness (FR5)	+	∼	-	+	+	+
Data confidentiality (FR6)	+	-	-	+	+	+
Data availability (FR7)	-	-	-	-	-	-
Embedded devices (DC1)	+	+	+	+	-	∼
Communication models (DC2)	-	∼	-	-	∼	-
Scalability (DC3)	-	-	-	-	-	-
Non IP networks (DC4)	+	+	+	+	-	+
QoS parameters (DC5)	-	∼	-	∼	∼	∼

Fig. 5. Evaluation of available standards

marizes the results of this security analysis.[2] At the left hand side, the BAS standards that have been described in Section 3 are listed. To show that using security mechanisms from other domains is not trivially possible, two of the most important IT security mechanisms were investigated, too. The corresponding results are shown at the right hand side of the figure.

4.1 BACnet Addendum g

BACnet Addendum g provides a solid base for securing BAS. However, the following aspects are missing or left open:

- *Authorization (FR2)*: The distribution of the keys is handled by the key server. The actual distribution to the devices predefines which devices are able to communicate with each other and which devices are excluded from a relationship. As a result, the assignment of the keys to the devices defines the devices' access rights and thus their authorization. Since this procedure is not specified by the standard, authorization has to be realized by the application.
- *Data origin authentication (FR4)*: Guaranteeing data origin authentication is only possible if a key is limited to two devices. If, for example, the `General-Network-Access` key or an `Application` key that is distributed to multiple devices are used, the sender cannot be identified in a secure manner.
- *Data freshness (FR5)*: The security mechanisms require the existence of (loosely) synchronized device clocks. Otherwise, data freshness cannot be guaranteed since the used mechanisms rely on timestamps.
- *Data availability (FR7)*: Mechanisms to protect against interruption attacks (e.g., Denial-of-Service (DoS) attacks) are not supported. Therefore, data availability cannot be guaranteed.

[2] "+" denotes that the requirement or challenge is (nearly) satisfied, "∼" means that it is only partly fulfilled, and "-" implies that the used mechanism is insecure.

- *Communication models (DC2)*: BACnet only provides support for the client/server model – exchanging data within groups is not supported.
- *Scalability (DC3)*: The use of a single key server introduces a single-point-of-failure. Therefore, a scheme based on multiple key servers is desirable. While the use of multiple key servers is possible, the realization of such a concept is not specified. Important details like synchronization of key servers and the selection of the key server to be used (especially in case of a faulty key server) are not discussed.
- *QoS parameters (DC5)*: Since multicast is not supported at all, reliability and ordering within in communication groups are not supported, too. For broadcast communication, QoS features cannot be specified.

4.2 LonWorks

LonTalk's security concept suffers the following security flaws [11, 15]:

- *Entity authentication (FR1)*: The used protocol only supports the verification of the sender's identity. The identity of the receiver cannot be checked. Furthermore, the challenge-response mechanism can only be initiated by the sender. A receiver does not have the opportunity to demand secured requests.
- *Authorization (FR2)*: Authorization is not supported since the same key is used for all LonTalk devices.
- *Date integrity (FR3) and freshness (FR5)*: The length of the used shared secret keys is limited to 48 bits which is too short to avoid brute force attacks. Additionally, only the data portion of the application layer is used as input for the hash calculation. Headers from other layers including the address information are not protected.
- *Data origin authentication (FR4)*: Each device can only use one authentication key. This means that all devices that want to communicate with each other must share the same secret key. As a result, data origin authentication cannot be guaranteed in networks with more than two members.
- *Data confidentiality (FR6)*: Disclosure of confidential data cannot be avoided, since the data is transmitted in clear text.
- *Data availability (FR7)*: There are no countermeasures that avoid an interruption of communication.
- *Communication models (DC2)*: The usage of the authentication protocol is restricted to acknowledged services. If an unacknowledged transmission mode is used, the identity of the sender cannot be verified.
- *Scalability (DC3)*: Using authenticated multicast, each receiver generates its own random number and sends it to the sender. As a result, the sender must respond to all receivers with an individual calculated hash value. If a communication group contains n members, the sender must calculate $(n - 1)$ hash values. Furthermore, it is not possible to establish communication sessions and so, it is always necessary to transmit four messages for secured requests.

– *QoS parameters (DC5)*: LonTalk provides support for acknowledged communication services. However, a defined ordering within multicast groups cannot be guaranteed.

While these security flaws are related to the standard challenge-response protocol of LonTalk, most of them also apply to the security mechanism of the IP tunnelling scheme of LonTalk. Instead of the used cryptographic algorithms, LonTalk/IP uses MD5. However, since MD5 is not collision resistant, it is insecure, too. Another difference is that data freshness is not guaranteed at all due to the absence of a nonce (e.g., random number). Therefore, the security mechanism of LonTalk/IP cannot be considered as an improvement.

4.3 KNX

Since KNX's access protection is very rudimentary, it does not provide the necessary mechanisms to guarantee a secure environment [15]:

– *Entity authentication (FR1)*: It is not provided since the identity of the receiver is not verified.
– *Authorization (FR2)*: The passwords are transmitted in clear text. If an adversary has access to the network, the adversary can simply intercept and retrieve the transmitted password to impersonate devices. Furthermore, the source address of a transmitted message can be spoofed very easily and so, an adversary can inject malicious messages without knowing the password.
– *Data integrity (FR3), data origin authentication (FR4), data freshness (FR5), data confidentiality (FR6)*: These objectives are not guaranteed at all.
– *Data availability (FR7)*: Interruption attacks cannot be avoided.
– *Communication models (DC2)*: The access protection mechanism cannot be applied to control data communication in KNX. An unauthorized use of these services cannot be avoided.
– *Scalability (DC3)*: KNX does not support mechanisms to manage, generate, and distribute passwords in a secure manner. Therefore, the passwords must be specified manually. It is up to the system administrator to guarantee that this configuration is performed in a physically secured environment. Furthermore, the single management tool called ETS needs to be used. ETS uses only one password for the whole installation. Hence, the rudimentary access protection scheme does not scale to large systems since compromising a single device discloses the password of all devices.
– *QoS parameters (DC5)*: KNX only provides acknowledged communication services for unicast communication. For multicast or broadcast communication, only unacknowledged end-to-end communication services are available. A defined ordering is also not possible for these services.

4.4 ZigBee

The security concept of ZigBee provides a solid base for secure communication. However, the following requirements and challenges are not satisfied:

- *Authorization (FR2)*: The smallest security context in ZigBee is a device. Using different secret keys for different user applications on a single device is not possible. Therefore, access control is only provided on a per-device basis.
- *Data integrity (FR3)*: The security services provided by IEEE 802.15.4 are not used by ZigBee. As a result, the data link header is not secured since ZigBee only protects the network and/or application layer parts of the messages. Furthermore, data link layer services like sending beacon frames and associate requests are not secured. Therefore, security threats that are dedicated to the data link header or to data link services cannot be avoided (e.g., re-routing of network traffic).
- *Data availability (FR7)*: Interruption threats are not considered in ZigBee. Especially the joining procedure is vulnerable to DoS attacks. The first part of the joining process (i.e., address assignment, synchronization with coordinator) is not secured since entity authentication is only provided afterwards.
- *Communication models (DC2)*: While ZigBee defines a multicast communication service, it is not clear how group communication is secured in ZigBee. It seems that the only possibility is to use the network key. However, a secure separation between different multicast groups is not possible if the network key is used. Furthermore, data origin authentication cannot be guaranteed. Link keys cannot be used to secure multicast communication, since link keys can only be shared between two devices.
- *Scalability (DC3)*: Key management is handled by a single trust center which may result in a single-point-of-failure. Furthermore, in wide-range networks, multiple hops may be necessary to reach the trust center. Therefore, a security concept based on multiple trust centers is desirable.
- *QoS parameters (DC5)*: ZigBee provides a mechanism to detect duplicates. Acknowledged communication services are only available for unicast communication – acknowledged multicast or broadcast services are not provided.

4.5 Security mechanisms for the IT domain

Due to the widespread use of the Internet, security has been a major research field in the IT world for years. Therefore, many well-established IT security mechanisms exist. If available BAS standards do not provide the necessary countermeasures against security attacks, an obvious solution would be the use of already existing security schemes from the IT world. Therefore, two of the most well-known IT security extension that may come into consideration for the BAS domain are presented. These are Internet Protocol Security (IPsec) [16] and Transport Layer Security (TLS) [17].

As shown in Fig. 5, both extensions provide a solid base for securing the communication. However, since both mechanisms are tailored towards the use within IP networks, the domain-specific challenges of BAS are not fully satisfied:

- *Data availability (FR7)*: Counteracting interruption attacks and thus guaranteeing data availability is out of the scope of both extensions.
- *Embedded devices (DC1)*: While special implementations of TLS are suitable for embedded devices [18], porting IPsec to embedded environments is not

easy to achieve. The main reasons are the introduced protocol overhead and the computational cost of the used cryptographic algorithms.
- *Communication models (DC2)*: Using IPsec with multicast is only possible with special implementations [19]. TLS cannot be used to secure communication within groups since it is dedicated to the client/server model.
- *Scalability (DC3)*: Both extensions demand an existing key server infrastructure that is used to manage the required security primitives. However, the exact implementation is not specified by the standards. Therefore, special implementations that scale to large systems are necessary.
- *Non IP networks (DC4)*: Due to its nature, IPsec is dedicated to the use for the IP protocol. Using it within field networks requires major changes in the current IPsec protocol.
- *QoS parameters (DC5)*: Since IPsec is located at OSI layer 3, guaranteeing reliability and a defined ordering of messages are left to higher protocol layers. TLS uses sequence numbers to detect missing messages or duplicates. However, providing a retransmission service or a defined ordering of messages are left to the other protocol layers, too.

5 Conclusion

As shown in this paper, available BAS solutions do not satisfy the demands of security-critical applications. While some technologies provide a solid base for a secure communication (e.g., BACnet, ZigBee), there are even communication standards where security is still neglected (e.g., LonWorks, KNX). To reduce this lack of security, a possible solution is to enhance available BAS technologies by integrating existing security concepts from other domains (e.g., from the IT world). However, due to the domain-specific challenges, mapping available security mechanism into the BAS domain is not trivially possible.

As a result, many important issues remain unsolved. As shown in Fig. 5, guaranteeing data availability is not provided by any solution. The main reason is that relying on cryptographic techniques does not fully counteract DoS attacks. Another major problem is that most security concepts are based on the use of a single key server. However, within large networks, such a single entity results in a single-point-of-failure. Finally, guaranteed QoS parameters like reliability or a defined ordering of messages are also not fully supported by available solutions. However, these features are of great interest for all-in-one BAS solutions especially if services from the safety domain need to be integrated, too [20].

While this paper analyzes the security features of available BAS standards, the development of new approaches and schemes that overcome the lack of security of current solutions is already under way. To achieve data availability, an advanced security concept based on organizational countermeasures that counteracts DoS attacks is presented in [21]. Furthermore, to eliminate a single-point-of-failure within the used secret key management protocols, two concepts that are based on multiple, redundant key servers have also been published [22, 23].

References

1. Pfleeger, C.P., Pfleeger, S.L.: Security in Computing, 4th ed. Prentice Hall (2006)
2. Dzung, D., Naedele, M., Hof, T.V., Crevatin, M.: Security for Industrial Communication Systems. In: Proceedings of the IEEE, vol. 93(6), pp. 1152 – 1177 (2005)
3. Perrig, A., Szewczyk, R., Tygar, J.D., Wen, V., Culler, D.E.: SPINS: Security Protocols for Sensor Networks. In: 7th Annual International Conference on Mobile Computing and Networking, pp. 189 – 199 (2001)
4. BACnet – A Data Communication Protocol for Building Automation and Control Networks, ANSI/ASHRAE 135-2008 (2008)
5. Building Automation and Control Systems (BACS) – Part 5: Data Communication Protocol. ISO 16484-5, (2007)
6. Control Network Protocol Specification. ANSI/EIA/CEA 709 Rev. B (2002)
7. Open Data Communication in Building Automation, Controls and Building Management – Control Network Protocol. ISO/IEC 14908 (2008)
8. Information Technology – Home Electronic Systems (HES) Architecture. ISO/IEC 14543-3 (2006)
9. KNX Specification Version 2.0, Konnex Association, Diegem (2009)
10. ZigBee Specification, ZigBee Alliance, San Ramon (2007)
11. Schwaiger, C., Treytl, A.: Smart Card Based Security for Fieldbus Systems. In: 9th IEEE Conference on Emerging Technologies and Factory Automation, vol. 1, pp. 398 – 406 (2003)
12. Holmberg, D. G.: BACnet Wide Area Network Security Threat Assessment. Technical report, National Institute of Standards and Technology, NISTIR 7009 (2003)
13. Zachary, J., Brooks, R., Thompson, D.: Secure Integration of Building Networks into the Global Internet. Technical report, National Institute of Standards and Technology, NIST GCR 02-837 (2002)
14. BACnet – A Data Communication Protocol for Building Automation and Control Networks. ANSI/ASHRAE 135-2008: Addendum g (2009)
15. Granzer, W., Kastner, W., Neugschwandtner, G., Praus, F.: Security in Networked Building Automation Systems. In: 6th IEEE International Workshop on Factory Communication Systems, pp. 283 – 292 (2006)
16. Kent, S., Seo, K.: Security Architecture for the Internet Protocol. RFC 4301 (2005)
17. Dierks, T, Rescorla, E.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (2008)
18. Gupta, V., Wurm, M., Zhu, Y., Millard, M., Fung, S., Gura, N., Eberle, H., Shantz, S.C: Sizzle: A Standards-based End-to-End Security Architecture for the Embedded Internet. In: Pervasive and Mobile Computing, vol. 1(4), pp. 425 – 445 (2005)
19. Weis, B., Gross, G., Ignjatic, D.: Multicast Extensions to the Security Architecture for the Internet Protocol. RFC 5374 (2008)
20. Kastner, W., Novak, T.: Functional Safety in Building Automation. In: 14th IEEE Conference on Emerging Technologies and Factory Automation, pp. 1 – 8 (2009)
21. Granzer, W., Reinisch, C., Kastner, W.: Denial-of-Service in Automation Systems. In: 13th IEEE Conference on Emerging Technologies and Factory Automation, pp. 468 – 471 (2008)
22. Granzer, W., Reinisch, C., Kastner, W.: Key Set Management in Networked Building Automation Systems using Multiple Key Servers. In: 7th IEEE International Workshop on Factory Communication Systems, pp. 205 – 214 (2008)
23. Granzer, W., Lechner, D., Praus, F., Kastner, W.: Securing IP Backbones in Building Automation Networks. In: 7th IEEE International Conference on Industrial Informatics, pp. 410 – 415 (2009)