# Increasing security and availability in KNX networks

Harald Glanzer[1], Lukas Krammer[1], Wolfgang Kastner[1]

**Abstract:** Buildings contain a number of technical systems in order to be able to fulfill their task of providing a comfortable, secure and safe environment. Apart from heating, ventilation and air-conditioning as well as lighting and shading, critical services such as fire alarm or access control systems are added to building automation. The latter services require secure communication and high availability and are currently implemented by isolated subsystems. However, a tighter integration into an overall building automation network can raise synergies such as cost reduction, improvements in building control as well as easier management. For this purpose, the underlying communication system has to be robust and reliable against malicious manipulations. This paper proposes an extension for KNX paving the way for its deployment even in critical environments. For this purpose, it is necessary to detect and guard against malicious attacks as well as to cope with randomly occurring hardware faults. The former can be achieved through cryptography, whereas the latter by implementing structural redundancy. The proposal divides KNX installations into insecure and secure parts. While insecure parts allow to use standard KNX devices, secure parts are protected against malicious attacks and are realized in a redundant way. This allows to partially resist against transient hardware faults.

**Keywords:** Building automation, Security, Availability, KNX, Smart building

## 1 Introduction

Building automation (BA) is a widespread topic that evolved over the past decades [Ka05]. Initially, BA systems were used for heating, ventilation and air-conditioning (HVAC) applications as well as for lighting and shading. Today, the term BA covers many more application domains such as alarm systems, access control or safety systems. Combining all these domains leads to ”intelligent buildings” and promises reduced maintenance costs, energy savings and improved user comfort compensating the primarily higher investment costs of such buildings. However, each application domain has different demands regarding dependability with the attributes reliability, availability, safety, confidentiality, integrity and maintainability [Av04].

In BA, the exchange of control data is a key issue. Small amounts of data are transmitted infrequently, but dependably over long distances. A number of control networks cater for this domain (e.g., BACnet, LonWorks, KNX, ZigBee, DALI). Integrating all application fields of BA mentioned above into an overall control network would, nevertheless, raise synergies in terms of sensor fusion and sensor sharing, as long as such a system could meet all dependability attributes.

In the early days of BA, this vision was contradicted by the fact that (communication) security was not considered as a critical requirement. The possible threats by misusing

[1] Technische Universität Wien, Automation Systems Group, contact author: k@auto.tuwien.ac.at

HVAC applications were assumed to be negligible. Additionally it was argued that the control networks were physically isolated. Finally, the underlying nodes were characterized by very limited processing power. Thus, the comprehensive use of cryptographic mechanisms would have put remarkable computing loads onto these nodes and was considered impracticable.

Meanwhile, system integration continued until a point where security concerns can no longer be neglected. BA systems and their networks are more and more linked together crossing former building borders for opening new application fields (e.g., for demand and load side management in smart grids). Considering such applications, a wide range of attacks has become possible. Intercepting and replaying datagrams allows an adversary to introduce arbitrary control data, for instance, to open doors or to disable HVAC systems without permission. Passive attackers can monitor the network traffic to analyze the types of active devices, gathering knowledge that can be used to develop further attack strategies. Denial-of-Service (DoS) attacks disabling building services can be conducted by simply physically shortcutting or interrupting a line connection, rendering the corresponding network segment unavailable. Such attacks must be precluded for sensitive services like fire or burglar alarm systems relying on the availability of the communication network. Availability, in general, can only be achieved by structural redundancy, i.e., by using replicated resources. Therefore, all resources needed for transmitting data between two points must exist redundantly and independently.

KNX is an open and widespread BA technology. It uses a layered structure and supports wired communication over Twisted Pair (TP) and Power Line (PL) as well as wireless communication by radio transmission. In addition, it supports communication with IP based hosts by a special type of router (KNXnet/IP). The present paper is focused on the design of a secure and highly available KNX network that also considers interoperability and backward compatibility, allowing the usage of KNX even in environments with increased safety-critical requirements. The proposed solution claims to be resistant against malicious adversaries as well as transient and permanent hardware faults. To achieve this, so called ”KNX security routers” are introduced. These devices possess two kinds of KNX interfaces. One kind of interface is connected to standard KNX networks. The second interface constitutes the entry point to a secured KNX network which is connected to the secured interfaces of other KNX security routers. To achieve higher availability, these secured interfaces and the corresponding communication lines must exist redundantly. This ensures that even in case of a DoS attack against one physical connection, communication is still possible.

## 2 Building automation networks and attack scenarios

Communication networks for BA are usually built upon a two-tier model consisting of a field level and a backbone level [Ka05]. The field level contains sensors, actuators and controllers interacting with the environment and performing the control functions. The field level devices are connected via fieldbus systems which in turn are coupled to a common backbone network via interconnection devices. The backbone network is home for man-

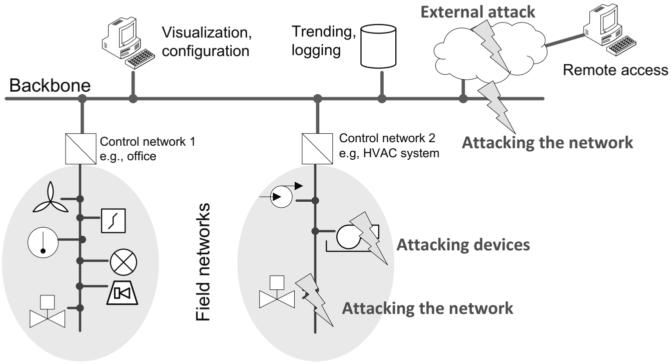

Fig. 1: Building automation networks and attack scenarios

agement devices which are used for configuration, visualization and monitoring. Special interconnection devices may act as gateways providing a connection to foreign networks (Figure 1).

Based on this topology, two different classes of attacks are possible: network attacks and device attacks [Gr10]. In case of network attacks, an adversary tries to compromise the field or backbone network. In the first scenario, the attacker can analyze or modify the control data of specific network segments. In the latter, the attacker gains access to the aggregated data and can thus obtain a global view of the system. To protect the communication, it would be possible to use security mechanisms well-known from the IT world. Unfortunately, these mechanisms cannot be mapped directly to BA networks because of the introduced overhead.

Alternatively, attacks against sensors, actuators or controllers to manipulate their behavior or attacks against interconnection devices to limit data access to and from the specific segments are possible. Finally, an adversary can launch an attack against the management devices to gain control over management applications (e.g., for accessing trends or logs). In general, such device attacks are divided into three categories: software attacks, side-channel and physical attacks, extensively surveyed in [GPK10] and [KS05].

## 3 KNX and KNX Security

KNX emerged from three leading standards namely European Installation Bus (EIB), European Home Systems Protocol (EHS) and Batibus. It is an open, platform independent standard, developed by the KNX Association implementing the ISO/IEC 14543-3 standard for home and building electronic systems. To provide platform independence, the standard uses a layered structure, based on the OSI model omitting layers 5 and 6. Different kinds of physical media are supported, allowing its use in different environments. Two different kinds of addresses are defined. The Individual Address (IA) of a KNX node is related to its position within the topological structure of the network. It specifies the number of the zone and (sub-)line the device resides in as well as its device number within the line. Group Addresses (GA) are logical identifiers and used for group communication. KNX supports interoperability between products from different manufacturers. This is achieved by an interworking model, which rests upon the concept of functional blocks and standardized data point types. For configuration and parametrization of the devices, the Engineering Tool Software (ETS) is used.

KNX does not implement security features except a rudimentary password-based control for management communication. The used keys are transmitted as cleartext, enabling an attacker to perform a passive attack to obtain the password. Subsequently, the attacker can mount an active attack, injecting arbitrary management messages. No methods are foreseen for generation or distribution of the keys. For control data, an adversary can directly inject arbitrary messages. These shortcomings disqualified KNX for usage in critical environments, restricting its possible fields of application. For this reason, a number of security extensions has been proposed (e.g., [Le09, CCM10]).

### 3.1 KNX Data Security

In 2013, the KNX Association published ”KNX Application Note 158” which specifies the KNX Secure Application Layer (S-AL) providing authentication and encryption, and the Application Interface Layer (AIL) implementing access control, both being part of the application layer. The settlement of these functions above the transport layer allows a transparent and media-independent end-to-end protection comprising of encryption and authentication [Kr13].

KNX S-AL services define modes for authenticated encryption or authentication-only of a higher-level cleartext Application Protocol Data Unit (APDU). This is achieved by combining Counter Mode AES-128 for encryption with Cipher Block Chaining (CBC) for generating the Message Authentication Code (MAC), also known as Counter with CBC-MAC (CCM). A critical parameter of CCM is the sequence number, a simple counter value that provides data freshness, thus preventing replay attacks, and is sent along with every KNX S-AL protocol data unit. This sequence number has to be synchronized by communicating devices. Since no sequence number can be used to guarantee data freshness at this stage, a challenge-response mechanism is used instead. For encrypting and authenticating KNX frames, different types of keys are used: a Factory Default Setup Key (FDSK) is used for initial setup with the ETS. The ETS then generates the Tool Key (TK) which is used by the device for securing of the outgoing messages. Consequently, every device must know the TK of its communication partners. While the S-AL empowers two devices to communicate in a secure way, the AIL allows a fine-grained control access control. Therefore, every link (a combination of source address and data or service object) is connected with a role, which in turn has some specific permissions.

### 3.2 KNX IP Secure

KNX IP Secure, as published in ”KNX Application Note 159”, is a security extension for KNXnet/IP that aims at being backward compatible. The KNXnet/IP traffic is encapsulated in KNXnet/IP Secure wrapper frames which should provide confidentiality, integrity, freshness and authenticity.

KNX IP Secure predominantly uses symmetric cryptography mechanisms. More precisely, it uses the AES-128 as basic block cipher for all modes of operation required in the specification. There are two types of communication in KNX IP Secure, namely unicast communication and multicast communication. In multicast communication, the traffic between members of one group should be secured. Thereby, it exclusively relies on symmetric cryptography schemes. Similar to KNX Data Security, CCM is used as mode of operation. There is also a special case in group communication. This communication type uses a pre-shared secret called Group Key. This key is unique for every IP multicast group. The same group key can be found on every device that is in the same group. A device can only be in one IP multicast group at a time. Unicast traffic is mainly used for configuration purposes, thus securing the communication between a management device and an interconnection device. To achieve perfect forward secrecy, KNX IP Secure uses Elliptic Curve Diffie Hellman (ECDH) key exchange algorithm over NIST curve K-283. Although the draft standard has been proposed recently, KNX IP Secure is still analyzed regarding its security properties (cf. [JKK14].

## 4 System Architecture

This section presents a KNX extension applicable to environments with increased availability demands for TP based KNX networks. The extension is designed in a transparent way utilizing a ”plug & play” functionality to build a secured KNX network [Gl15]. A standard KNX device outside this secured network should be able to send and receive messages via the secured network without any prerequisites. Every device with one connection to an unsecured KNX TP network and two distinct TP connections to a secured KNX network will act as a KNX security router. Thus, the presence of at least two of these KNX security routers connected to each other by two secured TP lines will constitute a secured KNX area spanning between installations with increased security demands as shown in Figure 2. The secured network is reserved for security routers only, i.e. no standard KNX devices are allowed here.

The basic tasks of the KNX security routers consist of

1. establishing keys with their communication partners within the secured KNX network,
2. providing redundant communication lines, achieving improved availability by encrypting and authenticating all messages which are received on the unsecured line, and delivering them to the proper KNX security router which acts as border device for the given GA, and
3. checking all messages which are received on the secured lines for integrity and authenticity, removing duplicates, unwrapping and delivering them to the unsecured line.

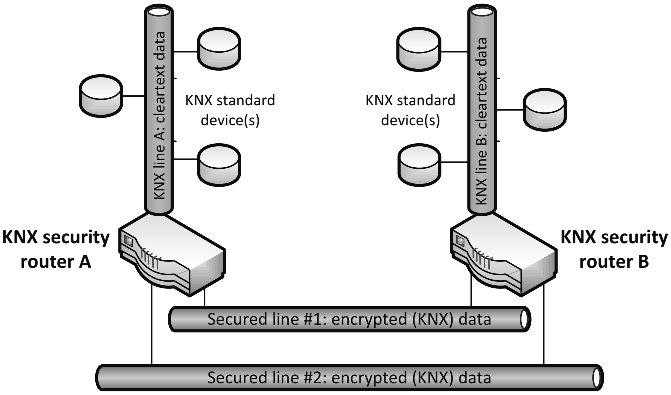
KNX line A: cleartext data
KNX standard device(s)
KNX standard device(s)
KNX line B: cleartext data
KNX security router A
KNX security router B
Secured line #1: encrypted (KNX) data
Secured line #2: encrypted (KNX) data

Fig. 2: Unsecured and secured KNX lines

In order to keep the communication overhead as small as possible, an efficient concept for translating KNX addresses to secure KNX addresses (and vice versa) is defined. When a new KNX security router is added to the network, it generates a list of group addresses it acts as border router for. GA out of this list can be addressed later on. This allows to dynamically deploy new KNX security routers with connected unsecured devices achieving a compromise between flexibility and complexity. Sending data to a GA follows the triad of discovery request, discovery response and data transfer, illustrated in Figure 3. Broadcast messages are depicted as solid end of the arrow, while others denote unicast messages.

To enable multiple devices to announce responsibility for a GA, a KNX security router in charge of forwarding data, must accept discovery responses following its own request message for a short time window. The discovery messages generated by KNX security routers have to be encrypted. Although these datagrams do not contain KNX payload per se, they allow an adversary to learn the topology of the network. For example, if an attacker detects that a particular KNX security router is responsible for only one GA and further gets knowledge that this GA is responsible for switching a light (i.e., by visual observation), the attacker afterwards may be able to derive a personal profile just by detecting packets for this GA, although the payload of the datagrams to the responsible KNX security router are encrypted (see below).

Discovery requests are broadcast messages readable by all KNX security routers. To limit the protocol overhead, a global network key is used. For providing authenticity, all datagrams passing the secured KNX network must contain a MAC to prevent modification of them. Defense against replay attacks is achieved by counters. The counters must be strictly monotonically increasing and must not overflow. They can be compared to an initialization vector that prevents the mapping of same cleartext messages to same ciphertext messages under the same encryption key.

Two different types of counters are used: one global counter $Ctr_{global}$ and individual counters $Ctr_{ind}$. The first one is used for avoiding replay attacks against discovery messages. $Ctr_{global}$ is a 4 byte integer, allowing $2^{32} \approx 4,3$ billion discovery request or response messages to be sent before overflowing. For synchronizing the global counter $Ctr_{global}$, a specific service is defined, allowing a newly powered-up device that wants to join the network to synchronize with the rest of the network. $Ctr_{ind}$ is used for the control data transfer. Beside avoiding replay attacks, this counter is necessary to detect and delete duplicates caused by the redundant network. Firstly, it is used as outgoing counter value $Ctr_{out}$ when sending control data from a specific IA. Secondly, every security router maintains a list of incoming individual counter values, $Ctr_{in}$, also referenced by the unique IAs, thus allowing to discard the duplicates.

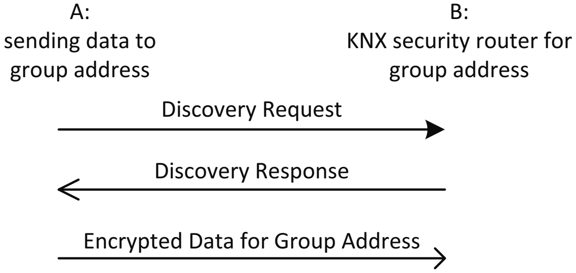

Fig. 3: Discovery and data communication phase

A feasible way to unambiguously identify duplicates is by referencing both $Ctr_{out}$ and $Ctr_{in}$ by the IA of the original cleartext message. This solution works despite potential network faults on one or both secured lines, provided that each KNX device is identified by a unique IA.

While it would be possible to use a centralized concept for the key management, no trusted on-line party is typically available in building automation networks. A centralized approach would need fall-back key servers which inherit the task of generating and distributing keys and parameters in case of a master key server failure. Otherwise, the network would suffer from a single point of failure in case no fall-back mechanism is applied – an assumption that would clearly disqualify the design as highly available. In contrast, the following lean key management is proposed:

- Known by all KNX security routers, a long-term key is defined. This pre-shared key $k_{psk}$ must be copied to every device at setup time and is used for symmetric encryption: (1) it authenticates synchronization messages; (2) it encrypts discovery requests and decrypts discovery responses; (3) it authenticates discovery messages.
- Another pre-shared key is used to authenticate the Diffie-Hellman (DH) parameters, as defined next.
- Asymmetric keys are used for end-to-end encryption of the actual data packets between two KNX security routers. ECDH serves as key negotiation algorithm. To protect against man-in-the-middle attacks, authenticity of the DH parameters must be assured.

A powered-up KNX security router must at first obtain the global counter $Ctr_{global}$ by sending out a synchronization request. This counter is used to avoid deterministic encryption of discovery messages. To authenticate the joining device and to avoid replay attacks a MAC in combination with a time stamp is used, which means that all KNX security routers must have their system clocks synchronized. To loosen this restriction, a time-window of deviation in the order of seconds is allowed.

If a KNX security router receives a cleartext message, it will at first check the counter value $Ctr_{out}$ for the IA and increment it. After that, the discovery phase takes place by sending request messages. These discovery requests are answered by KNX security routers with discovery responses. Discovery messages must contain the strictly monotonically increasing $Ctr_{global}$, the referenced GA and will also transport the chosen DH parameters used for later encryption of the actual KNX data frame. Authenticity is assured by a MAC. If at least one valid reply is received (correct MAC and incremented $Ctr_{global}$), the origin cleartext packet is duplicated, encrypted, put into a unicast data frame together with $Ctr_{out}$ and sent on both lines. The KNX security router in charge of delivering the frame to its KNX segment must maintain a counter for incoming packets ($Ctr_{in}$) which will be updated by the received counter value. An update occurs as soon as the first frame is received if $Ctr_{in}$ is higher than the stored counter. Subsequently, the delivery of duplicates can be detected because the received counter value will equal the saved counter value. Depending on the availability of the redundant channels, three cases can be distinguished:

- If both secured lines are available, one data frame will be handled first by the receiving KNX security router and the counter will be saved as actual $Ctr_{in}$ for the IA of the inner frame. When handling the second frame, the received counter will equal to the saved counter, and thus the frame will be discarded.
- If only one secured line is available, no duplicate will arrive, but the receiving KNX security router(s) will nevertheless update the received counter value for the IA.
- If both lines are unavailable, the responsible KNX security routers cannot update the corresponding value for $Ctr_{in}$. Nevertheless, the sending side will further update the outgoing counter $Ctr_{out}$ for the given IA. As soon as the responsible KNX security routers are reachable again, new data frames will contain a higher counter than the one saved on the receiving side, allowing data transfer to the GA again.

## 5 Discussion

The aim of this section it to show that the communication network is able to resist against maliciously introduced attacks as well as unintended faults happening randomly which results in a KNX network with improved availability. In the case of DoS attacks, the proposed solution can withstand such an attack against one of its two secured communication lines because of the applied redundancy. It is assumed that an attacker only has polynomially bounded processing power and is able to passively read frames from and inject arbitrary frames into both secured communication lines. In contrast, it is assumed that the cleartext KNX lines are out of reach of an attacker. Also, a KNX security router's hardware is expected to be physically secured, in particular the memory holding the long-term encryption and authentication keys.

### 5.1 Communication overhead

Neglecting the overhead for synchronization messages, one discovery request has to be sent for every cleartext KNX message answered by one response message for every KNX security router handling the GA in question. The exact size of these messages depends on the DH parameters. Additionally, the whole cleartext message (header and payload) will be encrypted and wrapped into another frame together with a MAC, $Ctr_{ind}$ and the header information of the outer frame. This message will be encrypted and sent independently to all KNX security routers that announced responsibility for the GA.

For a large number of KNX security routers responsible for a given GA, the proposed approach introduces a significant network overhead. This overhead could be reduced by using a GA cache. However such a cache was not considered in this proposal, yet.

### 5.2 Synchronization phase

A KNX security router joining the network must get knowledge of the actual value of $Ctr_{global}$. This is achieved by sending a broadcast message on every secured line serving as synchronization request. The frame contains the device's local time in seconds. Every device receiving such a request checks the integrity of the message first by recalculating the MAC. Afterwards, freshness is checked by comparing the supplied time with its local time. If the timing information equals the device's own local time, the device returns a synchronization response frame, containing its local time and the actual counter value.

- Passive attacks: Packets in the synchronization phase are not encrypted, allowing a passive adversary to learn the value of the global counter value $Ctr_{global}$. Nevertheless, this counter is only used to avoid deterministic encryption and is of no use for the attacker.
- Active attacks: An active attacker can inject new synchronization request and response messages, but will fail to produce a correct MAC for the actual time stamp with probability $1 - \frac{1}{2^{32}}$ because the MAC equals a random 32 bit number for the given header and payload. Such a MAC forgery will be detected by all active KNX security routers, and the corresponding frame will be discarded. Opening a window for tolerating clock deviations allows an attacker to resend captured synchronization messages within that time window. In case of a replayed synchronization request message, the attacker can trigger a new synchronization response message by a legitimate KNX security router. The response message will return the actual counter value to the original source address of the replayed message. The corresponding device however has already finished the synchronization phase and will just drop the message. When replaying a synchronization response message within the valid time window, there are two possibilities. If a joining device is waiting for a response message, the replayed message will be handled as legitimate response, and the newly joined device concludes the synchronization phase. On the other hand, if no device is waiting for a synchronization response, the message will simply be dropped.

### 5.3 Discovery phase

Whenever a KNX router receives a message on its unsecured line, two distinct discovery broadcast messages are sent, one for each secured line. Every device in the network first checks the authenticity of the received frame by recalculating the MAC. The requested GA is obtained by decrypting the corresponding field. Every KNX router acting as a border router for the GA prepares a response frame. During this phase, also ECDH parameter are exchanged. Integrity of the discovery messages is achieved by a MAC.

- Passive attacks: In this phase, a passive attacker is able to learn the global counter value $Ctr_{global}$ as well as the exchanged ECDH parameters. For the first case, the same arguments as given for the synchronization phase hold. For the second case, it can be argued as follows: to derive the key used by two parties in the subsequent data transmission from the DH parameters the attacker would need to solve the Elliptic Curve Discrete Logarithm Problem (ECDLP) [HMV04] – an infeasible task if proper ECDH parameters are chosen. Additionally, the attacker can learn the encrypted value of the requested GA and, thus, discovers which KNX security router(s) act as border routers. However, it is impossible for the attacker to derive the underlying cleartext GA because of the AES encryption, assuming that the attacker does not know the long-term encryption key. Nevertheless, the attacker is able to detect the basic topology of a network.
- Active attacks: For newly generated injected discovery request messages, the attacker must at first generate the encryption for the expected GA. Afterwards, the attacker must additionally guess the correct MAC. Trials to forge the correct MAC will be detected by all receiving devices. Similar arguments hold for discovery response messages. Replaying discovery messages is considered as uncritical because of the freshness property provided by $Ctr_{global}$. Such repeated frames will be detected by the KNX security router because of the outdated value of $Ctr_{global}$, which will just drop the replayed frame. Attacking alternating secured communication lines will also fail. For example, the attacker could at first shortcut one secured communication line such that the discovery message will not reach its recipient(s). Receiving the discovery message on the other line and injecting the frame to the previously blocked line would result in a fresh counter value. Alternating the source and/or destination addresses will invalidate the MAC, forcing the attacker again to forge the MAC, an infeasible task as already stated.

### 5.4 Data transmission phase

During the data transmission phase, a KNX security router in charge of forwarding a KNX message over the secured lines can also derive pairwise shared secrets with all responsible KNX security routers based on ECDH. From this shared secret, a key is derived which is used to encrypt the control data and inserted into the frame after the counter value.

- Passive attacks: An eavesdropping attacker will be able to learn source and destination addresses of the KNX security routers exchanging the frame, as well as the length of the original frame and the individual counter $Ctr_{ind}$. Again, the meta information can only be used to generate communication profiles, a fact considered inevitable. The individual counter is used as freshness property and to detect the duplicate frames on the receiving side. Therefore, an attacker does not benefit from knowing this counter value. Decrypting the contained inner frame is considered impossible based on the following facts: (1) encryption is based on AES-256, therefore trying all possible keys is infeasible. (2) The attacker is unable to get knowledge of the key because of the key agreement protocol used in the discovery phase.
- Active attacks: An attacker, trying to inject a new data frame, must succeed in forging the correct MAC. A MAC mismatch will be detected by the receiving KNX security router. A replayed message will be correctly verified and decrypted by the receiving device, but because of the outdated counter value $Ctr_{ind}$, the message will be discarded.

## 6 Conclusion and Outlook

In order to be able to deploy a communication system in more demanding environments, it is necessary to achieve informational security combined with mechanisms for increasing the availability of a system. KNX is a well-established control network technology tailored to the special needs of buildings. Originally, it was not designed for the use in critical environments. Increasing the security and especially the availability of KNX would allow its deployment also for critical applications. For KNX, extensions for securing a network against malicious attacks exist, but these extensions are not able to handle faults concerning the communication medium as well as DoS attacks. This paper work proposes a solution unifying all three columns of information security, namely availability, confidentiality and integrity, thus protecting against active and passive adversaries including even transient hardware faults. The proposal is even able to resist restricted DoS attacks. This is achieved by using KNX security routers which are connected to each other in a redundant way. A standard KNX line is connected through a KNX security router which copies the received KNX frames into two properly secured frames and sends them over both secured communication lines. The receiving KNX security router will check the incoming frames for modification, discard one of the two copies and forward the remaining one to the destined KNX line.

The proposed solution can withstand malicious attacks as well as transient hardware faults on one of the secured lines. It allows to connect standard KNX devices which are spatially divided in a secure manner, bridging over areas where malicious behavior cannot be ruled out. The approach was inspired by security mechanisms defined by the recent KNX security extensions. However, since network stacks covering these features are not available it was decided to rest the approach currently on its own security mechanisms. The feasibility of the solution was tested by a proof of concept. For the KNX security routers, RaspberryPis in combination with KNX-USB-dongles were used. To interface with KNX, the client library of eibd[2] was used providing functions for sending and receiving KNX frames. The needed cryptographic functions were built based on OpenSSL[3].

An enhancement for future developments would be to integrate a caching mechanism for the mapping of GA to KNX security routers and encryption keys. A KNX security router receiving data from a KNX device would send a discovery message once and cache the address(es) of the responsible KNX security routers(s), together with the corresponding encryption key(s). Subsequent data transfer can be executed without the need of the discovery phase reducing the bus load. Finally, the presented approach could also be applied to other building automation networks such as LonWorks with slight modifications.

## References

[Av04] Avizienis, A.; Laprie, J.-C.; Randell, B.; Landwehr, C.: Basic concepts and taxonomy of dependable and secure computing. Dependable and Secure Computing, IEEE Transactions on, 1(1):11–33, Jan 2004.

[CCM10] Cavalieri, S.; Cutuli, G.; Malgeri, M.: A study on security mechanisms in KNX-based home/building automation networks. In: Emerging Technologies and Factory Automation (ETFA), 2010 IEEE Conference on. pp. 1–4, 9 2010.

[Gl15] Glanzer, Harald: Highly available KNX networks. Master's thesis, TU Wien, 2015.

[GPK10] Granzer, W.; Praus, F.; Kastner, W.: Security in Building Automation Systems. Industrial Electronics, IEEE Transactions on, 57(11):3622–3630, Nov 2010.

[Gr10] Granzer, Wolfgang: Secure Communication in Home and Building Automation Systems. PhD thesis, TU Wien, 2010.

[HMV04] Hankerson, D; Menezes, A.; Vanstone, S.: Guide to Elliptic Curve Cryptopgraphy. Springer, New York, 2004.

[JKK14] Judmayer, A.; Krammer, L.; Kastner, W.: On the security of security extensions for IP-based KNX networks. In: Factory Communication Systems (WFCS), 2014 10th IEEE Workshop on. pp. 1–10, May 2014.

[Ka05] Kastner, W.; Neugschwandtner, G.; Soucek, S.; Newmann, H.M.: Communication Systems for Building Automation and Control. Proceedings of the IEEE, 93(6):1178–1203, 6 2005.

[Kr13] Krammer, Lukas; Bruyne, Steven De; Kastner, Wolfgang; Granzer, Wolfgang: Security Erweiterung für den KNX Standard. In: Tagungsband – innosecure 2013 – Kongress mit Ausstellung für Innovationen in den Sicherheitstechnologien Velbert Heiligenhaus. pp. 31–39, 9 2013. german.

[KS05] Koeune, Franois; Standaert, Franois-Xavier: A Tutorial on Physical Security and Side-Channel Attacks. In (Aldini, Alessandro; Gorrieri, Roberto; Martinelli, Fabio, eds): Foundations of Security Analysis and Design III, volume 3655 of Lecture Notes in Computer Science, pp. 78–108. Springer Berlin Heidelberg, 2005.

[Le09] Lechner, D.; Granzer, W.; Praus, F.; Kastner, W.: Securing IP Backbones in Building Automation Networks. In: Proc. 7th IEEE International Conference on Industrial Informatics (INDIN '09). pp. 410–415, 6 2009.

[2] https://www.auto.tuwien.ac.at/ mkoegler/index.php/eibd

[3] https://www.openssl.org/