FAKULTÄT
FÜR !NFORMATIK
Faculty of Informatics

Proposal

# Highly available KNX networks

## Building a Security Gateway for KNX networks with improved availability

| | |
|---|---|
| Advisor: | Ao.Univ.Prof.Dr. Wolfgang Kastner |
| Assistance: | Dipl. Ing. Lukas Krammer |
| Department: | Institute of Computer Aided Automation<br>Automation Systems Group |

Vienna, 25.8.2014

(Signature of Author)

(Signature of Advisor)

## Problem statement

KNX is an open communications protocol for building automation. It uses a layered structure and supports wired communication over twisted pair and power line as well wireless communication by radio or infrared transmission. Additionally, it defines gateways to the TCP/IP world. As such, it can be used for controlling traditional services like heating, ventilation and air conditioning(HVAC), but also for more sophisticated applications [1] like surveillance or fire alarm systems of buildings.
Given these potential applications, such a communications protocol can constitute a potential vulnerable point an adversary could exploit if there are no countermeasures deployed. Possible attacks range from DOS attacks by simply physically shortcutting a line connection to replay attacks or eavesdropping, interception, altering and injecting of arbitrary telegrams. The countermeasure providing integrity, confidentiality and authenticity consists of authentication between the sender and receiver of a message, and encryption of these messages, combined in a security scheme called 'Authenticated Encryption'.

Encryption uses block or stream ciphers to garble the cleartext message into a block respectively stream of pseudo-random binary zeros and ones. Without possessing the key, the encryption must not be reversible in feasible time to protect the secret message from unauthorized entities.

Authenticity, on the other side, takes blocks of input data and produces a short (compared to the data itself, which can consist of an arbitrarily large number of blocks) 'message authentication code' or MAC for this data, which will be sent along with the data. The MAC function is some kind of one-way-function and it must be 'difficult' to reverse the process, i.e. find parts or the whole block which was feed to the function to obtain the given MAC. This way, tampering of the sent data can be detected and the corresponding datagram will be discarded if the message was altered. Analog to encryption, it must be infeasible to find other MACs for the same message.
Availability, in general, can only be achieved by redundancy, by using replicated resources. Therefore, all resources needed for transmitting data between two points must exist redundantly and independently from each other.

This work's overall topic is to build a gateway for KNX which improves security and availability, allowing to use KNX in a security-critical environment. As will be shown, the security concept defined in the actual KNX Standard must be regarded as insufficient because the standard simply doesn't provide any encryption or authentication[2].

## Expected results

The final goal of this work is to build a prototype of a secure and high-available KNX network. This network will consist of raspberry pi single board computers, each connected to the secure part of the KNX network with 2 KNX-USB-dongles. Additionally, each raspberry pi is connected by another USB dongle to another KNX network in the standard, unsecured way. Therefore, the rasperry pi's are gateways between a secure and an insecure KNX network, each of them running a master daemon responsible for reading datagrams from the KNX insecure world, encrypting them and sending them over the secure KNX lines. On the counterparts, the message will be decrypted, checked for integrity and sent on its further way if the message is found to be untampered.
While it is not obvious that confidentiality cannot be achieved without authenticity, it will be

shown that booth concepts must go hand in hand to achieve confidentiality. Consequently, the security layer will use strong encryption *and* authentication to guard against attacks.

For gaining availability, it would be possible to define some kind of 'heartbeat' protocol which checks the status of the 2 lines, and switches the line in case of a failure. A disadvantage of this approach is that a considerable amount of bus load is generated. Alternatively, it is possible to just copy every KNX message to booth lines (after encrypting it), and use some kind of counter to remove duplicates on the receiving side. This solution needs no additional traffic, and is also 'faster' because no fail over recognition time is needed.

Because a major concern of this work is, given the low bandwidth of KNX TP, to keep the introduced overhead as small as possible, the latter solution is preferred. It is important to note that the practical part of this work will only handle the twisted-pair media of KNX, although the basic principles can be deployed in a modified manner in wireless and power-line networks as well.

After the master daemon has been implemented, a threat analysis will be conducted to proof that the system can withstand the defined attacks and is robust, i.e. that it can recover from erroneus states. This will be done by building a small test setup, consisting of at least 2 security gateways and various KNX hardware, that will be exposed to the various defined attacks.

## Methodological approach

The methodological approach will follow the typical ordering, namely research, concept, implementation and evaluation / critical reflection:

Every secure system will just work within some defined barriers - it is impossible to build a system that is secure under all circumstances. So, the very first step will be to define a realistic thread scenario. Ensuring security in a network is a complex and comprehensive topic, fortunately, canonical ways how to employ authenticated encryption exists, therefore the next step is to decide which ciphers should be used, how the keys will be distributed and what kind of MAC to use. Additionally, a reasonable concept has to be found to guarantee high availability without introducing too much overhead. After it has been examined what kind of crypto-primitives will be used, the basic structure of the security layer must be defined. This layer should be as transparent as possible to allow it's re-use in different environments, i.e. different transport media. On top of the security layer, the methods to achieve availability have to be defined. As basis for implementation of a real-world testing environment, rasperry pi's will be used. These are small but powerful single board computers, based on ARM processors, using a Linux kernel as operating system. A C++ KNX API named 'eibd' exists and will be used for the KNX/USB interface. So, the resulting daemon will consist of a stack composed of EIBD, the security / key exchange layer in combination with a cryptographic library and the availability layer.

After implementation, it must be shown that this construction indeed withstands the defined attacks, and that the protocol works in practice.

## Structure of the work

1. Introduction
2. KNX

- Specification
- Overview of the defined layers
- Attacks on KNX

3. State of the art

   - Computer Security
     - Definition of Security
     - Types of Attacks
     - Basic Concepts of Number Theory used(Fermat, Euler)
     - Basic Concepts of Probability Theory used
     - Encryption Schemes
     - Authentication Schemes
     - Authenticated Encryption
     - Distribution of Keys
     - Symmetric vs. Asymmetric Encryption
     - Random Number Generators
   - Availability
     - Requirements for Availability
     - Protocol Used
     - Resynchronization

4. Security Concept

   - Key Generation and Distribution
   - Definition of the Status Messages
   - Definition of the MAC
   - Definition of the Cipher
   - Definition of the Tunneling Mode
   - Evaluation
     - Implemenation
     - Performance
     - Threat analysis

5. Critical Reflections

## State of the art

As stated earlier, KNX defines no methods for securing datagrams in the original proposal. On the other hand, some proposals which promise to fix this issue exist, nevertheless these projects didn't make it further than any theoretical level:

- eibsec[8] defines a security extension to KNX which uses AES encryption and dedicated key servers

- Salvatore Cavalieri and Giovanni Cutuli[4] propose another way how to authenticate and encrypt KNX traffic.

While these 2 approaches try to bring confidentiality to KNX, this work will also add availability to KNX, as stated above. Additionally, this work will not rely on key servers, thus eliminating the need for a single 'point of trust'.
In contrast to these proposals, this work will provide a real-world implementation, by using a tunneling method, similar to tunnel mode implemented by IPsec [10]. IPsec, an extension to IP, can authenticate and encrypt data sent with IP by defining two security services, namely the 'Authentication Header' to provide authenticity, and the 'Encapsulating Security Payload' for confidentiality. IKE, Internet Key Exchange, is used as key negotiating protocol. This way, IPsec can provide end-to-end encryption and protect the payload of higher level protocols like TCP or UDP.
Instead of self-implementing the needed crypto-primitives, the use of an API like 'crypto++' or 'Keyczar' is favored because these libraries are widely used, open source, maintained and have proven to be secure in the field. These libraries implement a wide range of authentication, encryption and key exchange modes, thus providing great flexibility. Therefore, some basic design decisions have to be made.

A fundamental property is the data format, i.e. if the data is to be handled in form of blocks or in form of a continuous data stream. Stream ciphers can be provable 'perfect secret' in principle and can be implemented as simple as bitwise xor'ing of key and data(i.e. the one-time pad).

Block ciphers come in 2 flavors: pseudo-random functions(PRF), and pseudo-random permutations(PRP). While the latter one is reversible, this is not true for the first one. Therefor, PRFs can only be used in constructions which do not depend on a reverse function, for example like 'Feistel Networks'. A widely used encryption standard is 'AES', the advanced encryption standard, derived from a block cipher called 'Rijndel'. This construction is reversible(i.e. is a PRP) and is also called a 'substitution-permutation-network', named after it's 2 basic building blocks.

Another important distinguishing point is the mode of operation: this basically means which construct is used to transform cleartext data into the needed pseudo-random representation, which underlying cipher is used, and also defines how this transformation is reversed for decryption. There are modes for encryption and authentication, some modes can also be used for booth tasks(i.e. cipher block chaining, CBC).

Additionally, this property decides what is done first: encryption, and afterwards authentication of the encrypted data, or authentication of the cleartext data, appending the obtained MAC to the cleartext data, and encrypting data and MAC afterwards. Depending on this ordering and what modes are used for authentication and encryption, this option may enable attacks like 'padding oracles'[12]. Another possibility is to generate a MAC for the cleartext data, and only encrypt the data itself. Obviously, care must be taken that the MAC does not carry any information about the cleartext data.

Another very fundamental difference is which kind of keys are used, i.e. symmetric vs. asymmetric encryption: while symmetric encryption is superior in regards of performance, symmetric keys need an already established, secure channel for key exchange. Because of that, a mixture of booth modes is used in prominent protocols(i.e. SSL[5]). The key exchange algorithm defines how the keys get distributed to all devices which are part of the secure network. It would be possible to fix a key, place this key on all devices and use this 'pre-shared secret' for symmetric encryption, but this key cannot be changed at a later point without direct interaction on all devices. A better way is to use some kind of asymmetric key which is used to establish session keys, which can be used in a symmetric manner. Well known examples of asymmetric or public key algorithms are 'RSA'[11] and 'Diffie-Helmann'[6]. While the latter one was originally based on exponentiation, a new method based on elliptic curves has been found which can achieve the same level of security with shorter keys. Depending on the mode of operation and the length of the used keys, there exists an upper bound on how many messages can be sent securely without changing the key. Therefore, it must be either made sure that this number cannot be achieved in a reasonable time, or some kind of key-renegotiation has to be used, which is the task of the key management algorithm. Nevertheless, if a session key and the corresponding traffic gets known to an adversary, all the past data will be disclosed(and all future traffic if no new key is used). Protocols like 'off the record messaging', OTR[3], avoid this problem by using short term session keys, and thus providing 'perfect forward secrecy'. This property ensures that, even if a key is known to an adversary, no future and no past messages can be decrypted(beside of one single message). Problematic about OTR is its lack of support for multi-party conversations, a feature that is tried to be achieved with multi-party OTR (mpOTR)[7][9].

Finally, a critical prerequisite is the used pseudo random number generator('PRNG'). While the used platform contains a hardware based PRNG, is has to be examined if the provided entropy can be considered secure, because a predictable PRNG turns all other cryptographic measurements useless.

## Relatedness to Computer Engineering

Modern cryptography relies heavily on number theory and probabilistic theory and is the basis of this work. The practical work will be to implement the multi-threaded daemons on the raspberry pi's, written in the low-level programming language C, by using the help of the C++ API 'eibd'.

Related lectures:

- 104.271 VO Discrete Mathematics
- 104.272 UE Discrete Mathematics
- 184.189 VU Cryptography
- 182.721 VO Embedded Systems Engineering
- 182.722 LU Embedded Systems Engineering
- 389.166 VU Signal Processing 1
- 183.624 VU Home and Building Automation

# Bibliography

[1] KNX Association. “KNX Applications”. URL: http://www.knx.org/fileadmin/downloads/08%20-%20KNX%20Flyers/KNX%20Solutions/KNX_Solutions_English.pdf.

[2] KNX Association. “System Specifications, Overview”. URL: http://www.knx.org/knx-en/knx/technology/specifications/index.php.

[3] N. Borisov, I. Goldberg, and E. Brewer. “Off-the-record Communication, or, Why Not to Use PGP”. In: *Proceedings of the 2004 ACM Workshop on Privacy in the Electronic Society*. Washington DC, USA: ACM, 2004, pp. 77–84.

[4] S. Cavalieri and G. Cutuli. “Implementing encryption and authentication in KNX using Diffie-Hellman and AES algorithms”. In: *Industrial Electronics, 2009. IECON ’09. 35th Annual Conference of IEEE*. 2009, pp. 2459–2464.

[5] T. Dierks and C. Allen. *The TLS Protocol Version 1.0*. RFC 2246. RFC Editor, 1999, pp. 1–80. URL: https://www.ietf.org/rfc/rfc2246.txt.

[6] W. Diffie and M.E. Hellman. “New directions in cryptography”. In: *Information Theory, IEEE Transactions on* 22.6 (1976), pp. 644–654.

[7] I. Goldberg et al. “Multi-party Off-the-record Messaging”. In: *Proceedings of the 16th ACM Conference on Computer and Communications Security*. CCS ’09. Chicago, Illinois, USA: ACM, 2009, pp. 358–368.

[8] W Granzer, G. Neugschwandtner, and W. Kastner. “EIBsec: A Security Extension to KNX/EIB”. In: *Konnex Scientific Conference*. Nov. 2006.

[9] L. Hong, E. Y. Vasserman, and N. Hopper. “Improved Group Off-the-record Messaging”. In: *Proceedings of the 12th ACM Workshop on Workshop on Privacy in the Electronic Society*. WPES ’13. Berlin, Germany: ACM, 2013, pp. 249–254.

[10] S. Kent and K. Seo. *Security Architecture for the Internet Protocol*. RFC 4301. RFC Editor, 2005, pp. 1–101. URL: http://tools.ietf.org/html/rfc4301.

[11] R. L. Rivest, A. Shamir, and L. Adleman. “A Method for Obtaining Digital Signatures and Public-key Cryptosystems”. In: *Commun. ACM* 21.2 (Feb. 1978), pp. 120–126.

[12] S. Vaudenay. “Security Flaws Induced by CBC Padding - Applications to SSL, IPSEC, WTLS”. In: *Proceedings of In Advances in Cryptology*. Springer-Verlag, 2002, pp. 534–546.