Secure Key Management in Embedded Systems: A First Proposal

Gorazd Jank, Silvia Schmidt and Manuel Koschuch[a]

Competence Centre for IT Security, University of Applied Sciences FH Campus Wien, Vienna, Austria

Keywords: Constrained Devices, Decision-making, Embedded Systems, Internet of Things, Secure Key Management.

Abstract: The Internet-of-Things (IoT) domain is highly heterogeneous and comprises a multitude of different devices. Because of this variety, many projects require unique compositions of tools, systems, and use cases. In addition, embedded devices are highly optimized and due to that are subject to different constraints. The interconnection of such products for data analysis or cooperation simultaneously increases the attack surface, which leads to requiring efficient cryptographic methods for the protection of data and communication. To enable this, a secure key management approach is needed. In practice however, there are still difficulties regarding the implementation and associated decision making of said management. All the more so since a generic one-size-fits-all approach in such a complex heterogeneous environment as the IoT simply does not exist. This paper aims to provide initial guidelines to argue the choice of a secure key management approach. To do so the state-of-the-art is presented and benefits as well as limits are evaluated. After that a set of factors and a first taxonomy are presented, which influence the final key management solution.

1 INTRODUCTION

The number of embedded systems continues to grow. Various summaries of market research suggest that this trend will continue in the next years (Wadhwani and Yadav, 2020). By the end of 2019 the market size exceeded USD 100 Billion and is expected to grow up to USD 160 billion by 2026.

Despite this widespread usage, a 2018 survey conducted by the Barr Group with over 1,700 participants concluded that *"About 1 in 6 designers of potentially injurious, Internet-connected embedded systems are completely ignoring security."* (Barr Group, 2018). This situation is exacerbated by the emergence of malware like Mirai (Kolias et al., 2017), which was first discovered in mid-2016. The worm compromised thousands of systems because basic security practices were ignored. The infected devices fell victim to only 62 default username and password combinations. Even a year later, not only did Mirai still infect IoT devices, but a vast amount of similar worms emerged, infecting even more systems (Kolias et al., 2017). This example and similar attacks (Nie et al., 2017; Checkpoint, 2018; Rajendran and Nivash, 2019) highlight the importance of IT/OT – Security in the IoT.

The goal of this work, based mainly on the results of the primary author's master's thesis (Jank, 2020), is to present a structured way to support the decision-making process of picking a suitable key management approach during early development of an embedded (IoT) system. The challenge for the decision makers lies within the heterogeneity of embedded devices, their individual constraints as well as the vast amount of possible key management compositions and the resulting advantages and limitations for the final system. Many factors have to be taken into account such as the architecture of the embedded systems, their security requirements and special threats. Furthermore, production, commissioning and maintenance costs as well as existing in-house know-how and infrastructure must be considered.

Numerous security measures such as user authentication, secure communication and data encryption require cryptographically secure keys to work correctly. In order not to compromise the algorithms and procedures, keys need to be generated, distributed, protected and destructed in a secure way. The process of performing these steps in a structured fashion is called key management (Barker, 2020).

Requirements of embedded systems differ from classic IT-systems such as PC or server systems. They often have restricted resources, longer product life cycles and a different focus on security goals (Sadeghi et al., 2015). This and the heterogeneous landscape

[a] https://orcid.org/0000-0001-8090-3784

Jank, G., Schmidt, S. and Koschuch, M.
Secure Key Management in Embedded Systems: A First Proposal.
DOI: 10.5220/0010475600810091
In *Proceedings of the 6th International Conference on Internet of Things, Big Data and Security (IoTBDS 2021)*, pages 81-91
ISBN: 978-989-758-504-3

make it difficult if not impossible to implement a generic security solution (Humayed et al., 2017).

Security in embedded systems is often reduced to an afterthought, while it should be considered starting from the design throughout the entire life cycle (Mirjalili and Lenstra, 2008). An embedded device in an industrial plant with an estimated lifetime of 20 years has completely different security requirements compared to a consumer IoT-Gadget. Considering security only at the end of the development limits the amount of suitable options.

The majority of existing research focuses on how to add to security while in the requirements and design phase (e.g. (M.K et al., 2016; European Union and Agency for Network and Information Security, 2017; European Union Agency for Cybersecurity, 2019)). These phases should output necessary security requirements and define processes on how to enforce them. During implementation, different security mechanisms will be needed to safeguard different parts of the solution. Those often need secrets, which in most cases will be realized through cryptographic keys. The assumption that a single key will remain secure for the entire lifetime of the embedded system typically is not sufficient (Grand View Research, 2014). To account for revocation of compromised keys and generation of new ones, key management solutions are required. This has however to be adapted to the specific needs of the embedded system that is using it (Grand View Research, 2014).

Various possibilities exist to implement key management. These differ by the usage of special hardware or software for calculations, by the utilization of symmetric or asymmetric keys, by the requirements on storage space, memory or CPU and other details (Vai et al., 2015; Whelihan et al., 2016; Obermaier et al., 2018).

This work aims to present a first introduction to our structured approach to support decision makers in the process of choosing a suitable, secure key management solution. The remainder of this paper is now structured as follows: Section 2 gives a comprehensive overview of existing works dealing with the problem of key management in constrained devices. From that, we present our contributions in form of a taxonomy to support decision making in Section 3 and finally concluding with a brief outlook in Section 4.

2 RELATED WORK

In the field of key management for embedded systems we are not aware of works supporting the decision making for all key management objectives. However, there is some work giving advice for classical key management and a paper presenting a method to pick a fitting key management protocol for the domain of Wireless Sensor Networks (WSNs).

Basic key management concepts are already defined in 1997's *Handbook of Applied Cryptography* by Menezes et al. (Menezes et al., 1997). This includes the key life cycle, the concept of Trusted Third Parties (TTPs), digital signatures, key protocols as well as fundamental advantages and disadvantages of different key management approaches.

The NIST publication "*Recommendation for key management: Part 2*" *[SP-800-57-2]* (Barker and Barker, 2019) contains a chapter handling the planning of key management. They start by choosing a fitting key management architecture based on the available cryptographic mechanism and objectives. After that a key management specification is developed for each cryptographic product used in the system. This specification has to list, for each contained device, information about key generation and/or distribution, key storage, access control, accounting and auditing, recovery from compromise, corruption, or loss of keying material as well as key recovery from backups and archives. In the third step a *Cryptographic Key Management System (CKMS) Security Policy* is developed. This is "*a set of rules that are established to describe the goals, responsibilities, and overall requirements for the management of cryptographic keying material throughout the entire key lifecycle*" (Barker and Barker, 2019). Finally, a decision is made if the CKMS is operated in-house or by another organization and a *CKMS Practice Statement* is developed. It describes the establishment of a *trust root* for the CKMS and a specification "*how key management procedures and techniques are used to enforce the CKMS Security Policy and to conform with the Key Management Specification*" (Barker and Barker, 2019). This planning procedure does not consider possible limitations which embedded devices can have.

Alcaraz et al. (Alcaraz et al., 2012) present an approach on how to map WSN requirements to key management protocol properties. They defined a set of attributes which they claim to be crucial when selecting a protocol. These attributes can be used to evaluate a set of candidates and pick one by excluding protocols not fitting the application requirements.

The authors of (Alohali et al., 2015) surveyed key management schemes for the *Smart Grid*. In the first part of their paper they discuss components of the smart grid and in the second part key management. They separate the schemes into key management for AMI, SCADA, V2G as well as WSN. Further approaches are separated into symmetric and asymmetric. The writers conclude that approaches for WSN "*can be effectively implemented for Smart Grid use*", that each analyzed scheme can be efficient depending on the setting and that there is no generic solution fitting for each Smart Grid application.

Messai et al. focus their survey on a Multi-Phase WSN (Messai and Seba, 2016). As WSN devices mostly work on battery, they have to be replaced from time to time. When a set of nodes is replaced after a fixed time this is called a *phase*. Those phases generate new challenges as nodes have to frequently leave and join the network. They analyzed the work of previous surveys on WSN but only take into account approaches fitting Multi-Phase WSN requirements. They classify them into deterministic and probabilistic approaches and compare their security and efficiency. Finally, they point out future open issues: the authors believe that addressing sensor node failure is of importance and for that fault-tolerant key management is needed. Furthermore they see a need for research into Mobile Multi-Phase WSN, the IoT as well as *Real-world applications*, because as yet there exist no implementations of the covered schemes.

In (Barskar and Chawla, 2016) the authors survey efficient group key management schemes in wireless networks. They separate the analyzed schemes into *Network Independent Schemes* and *Network Dependent Schemes*. They focus on network independent schemes which do not consider the underlying network infrastructure because they can also be used in wired and wireless settings. They further divide them into *Centralized*, *Decentralized* and *Distributed* group key management schemes. They examine each scheme for security requirements like forward and backward secrecy, and QoS requirements, e.g. low bandwidth overhead or minimal delays. They also summarize advantages and limitations of each subgroup.

Lavanya et al. surveyed key management in IoT (Lavanya and Usha, 2017). Ten key management techniques are covered which according to the authors are suitable for the IoT. The paper focuses on protocols rather than holistic approaches. They categorize key management into *public key cryptography*, *pre-shared key approaches* and *link-layer oriented* schemes. They conclude that various key management schemes for end-to-end security in high layers exist, but key management on network level needs further research. Further energy cost of establishment schemes in heterogeneous IoT networks have to be decreased.

Delay tolerant networks (DTNs) are another subgroup of embedded systems. Those systems aim to address technical issues with "*intermittent connectivity, network heterogeneity, and large delays*" (Menesidou et al., 2017). The authors of (Menesidou et al., 2017) count deep space, sensor-based and terrestrial wireless networks among them. They separate key management schemes into *Security Initialization*, *Key Establishment* and *Key Revocation*. Plenty of key management schemes are analyzed and the possible use of PKI and IBC in DTN is discussed. A conclusion of the paper is that "*hardware testbeds and real-life deployments in cryptographic key management are still largely missing from the DTN research area*". In the last part the authors also propose different key management categorization attempts e.g. if an approach needs a TTP or not. Also open research challenges are presented, highlighting shortcomings in most parts of DTN key management (e.g. key storage and key revocation) as well as deficiencies in DTN network research, for instance naming and performance issues.

Vasukidevi et al. present a paper reviewing existing key management and authentication techniques available in Vehicular Ad-Hoc Metworks (VANETs) (Vasukidevi and Sethukarasi, 2017). They try to take into account the difficulties of such networks and present a list of approaches with advantages and performance metrics. They are of the opinion that authentication plays a major role in securing VANET.

The writers of (Huang and Chen, 2018) published a survey about *key management service in cloud*. They analyzed *AWS CloudHSM*, *Keyless SSL* and *STYX* and focused on security and performance. They present the architecture of each approach and conclude that distributed environments of cloud platforms make key management more complicated.

Manikandan et al. made a survey on various key management schemes in WSN (Manikandan and Sakthi, 2018). They surveyed 13 schemes and separated them into *symmetric*, *asymmetric* and *hybrid* approaches. Furthermore, they analyzed these schemes with regard to computational complexity, memory, energy consumption, scalability, communication overhead, used technique and possible attacks. Their goal was to "*be helpful for researchers in selecting the appropriate key management technique specific to their application needs.*"

A paper surveying *key management for beyond 5G Mobile Small Cells* was published by De Ree et al. (De Ree et al., 2019). They discuss key management schemes for Mobile Ad-Hoc Networks (MANETs), as well as *ad hoc D2D networks*. The paper solely focuses on *self-organized* key management schemes. These schemes do not rely on an online centralized TTP. The authors classify approaches into *Public Key Cryptography* and *Symmetric Key Cryptography*. Further they subclassify *Public Key Cryptography* into *Certificate-based*, *Identity-based* and *Certificateless* approaches. They also defined some requirements of self-organized key management for networks of mobile small cells: security, connectivity, overhead, scalability, sustainability, fairness and secure routing independence. These requirements were then used to analyze the 17 approaches they looked into. The writers argue that there are two open research challenges. Firstly, key management schemes relying on a partially distributed TTP require a rigorous procedure for selecting the most suitable network nodes to act as the distributed TTP. Thereby it can happen that nodes act selfishly due to the overhead burden. This selfishness, as well as the many considerations needed in the selection process, need further attention. Secondly, the MANET schemes taken into account rely on physical contact to instantiate trust and distribute keys. The reason for that is the lack of network infrastructure. Such an approach is not fitting for mobile small cells, and for that network nodes must use online network infrastructure. There have been proposed schemes to secure D2D communication but they always assume that the network infrastructure is secure against compromise. Because of that the authors propose further research for authentication schemes *which prevent distribution of sensitive and private data over insecure channels*.

From all these works it can be seen that still no structured approach for supporting decision makers with the question, which approach or combination of solutions is best suited for their product, exists. The purpose of this paper is to propose a first attempt at such an approach.

3 KEY MANAGEMENT: A TAXONOMY

Commonly key management is classified by the employed underlying cryptographic primitive into symmetric and asymmetric (Wan et al., 2016; Manikandan and Sakthi, 2018; Payment Security Support Group, 2018; Barker, 2020; Malik et al., 2019) or by the resulting architecture into centralized, decentralized and distributed (Rafaeli and Hutchison, 2003; Challal and Seba, 2005; Mapoka, 2013; Abouhogail, 2014; Sharma and Krishna, 2015; Liu et al., 2020) approaches. Less common grouping criteria are separation by organization type (self-, TTP-organized) (De Ree et al., 2019), utilized communication patterns (peer-to-peer, group, mixed) (Kandi et al., 2020), key establishment approach (probabilistic, deterministic) (Messai and Seba, 2016), inclusion of biometric data (biometric, non-biometric) (Masdari et al., 2017), network topology (hierarchical, flat), network dependability (network dependent, network independent) (Mapoka, 2013) etc. An overview can be found in Figure 1.

These classifications may be used to characterize specific approaches. The approach in (Nafi et al., 2020) can for example be outlined as hybrid, decentralized, network independent, self-organized and non-biometric approach. Each class itself has advantages and disadvantages depending on the use case. This section introduces and analyzes those classes and highlights benefits and limitations when used with embedded systems.

3.1 Cryptographic Primitive

Approaches can be differentiated by the used cryptographic primitives (Barker, 2020). They can rely on symmetric or asymmetric cryptography, or utilize both primitives in which case they are called hybrid (Manikandan and Sakthi, 2018). In the context of this work hybrid approaches are those that combine the management of symmetric and asymmetric keys e.g. (Balasubramanian et al., 2005; Dawson et al., 2006). In (Balasubramanian et al., 2005) a decentralized approach is implemented using clusters. Within each cluster, symmetric keys are used. Inter-cluster communication is secured using asymmetric keys.

Not counting as hybrid approaches are those utilizing symmetric keys exclusively on protocol level. Asymmetric approaches often use protocols which use asymmetric cryptography to establish a secure symmetric session key with a communication partner. Traffic is then encrypted with symmetric algorithms to boost computational and energy efficiency (Saravanan et al., 2011). A common effective way to establish symmetric session keys is the ECDH protocol (Abusukhon et al., 2019). The key is exclusively used for one or a part of a session and is only saved on volatile storage.

To assure the same amount of security, asymmetric approaches need longer keys than symmetric approaches. A symmetric AES-128 key has a length of 128 bits while asymmetric RSA keys with a comparable security strength need 3,072 bits (Barker, 2020). As a consequence more computing power and storage is required (Haque et al., 2018). This effect can be reduced when using an ECC algorithm. To achieve same security strength as AES-128, the asymmetric ECDSA algorithm needs about 283 bits (Barker, 2020), which is only a fraction of RSA. A detailed comparison of ECC and RSA can be found in (Jansma and Arrendondo, 2004).

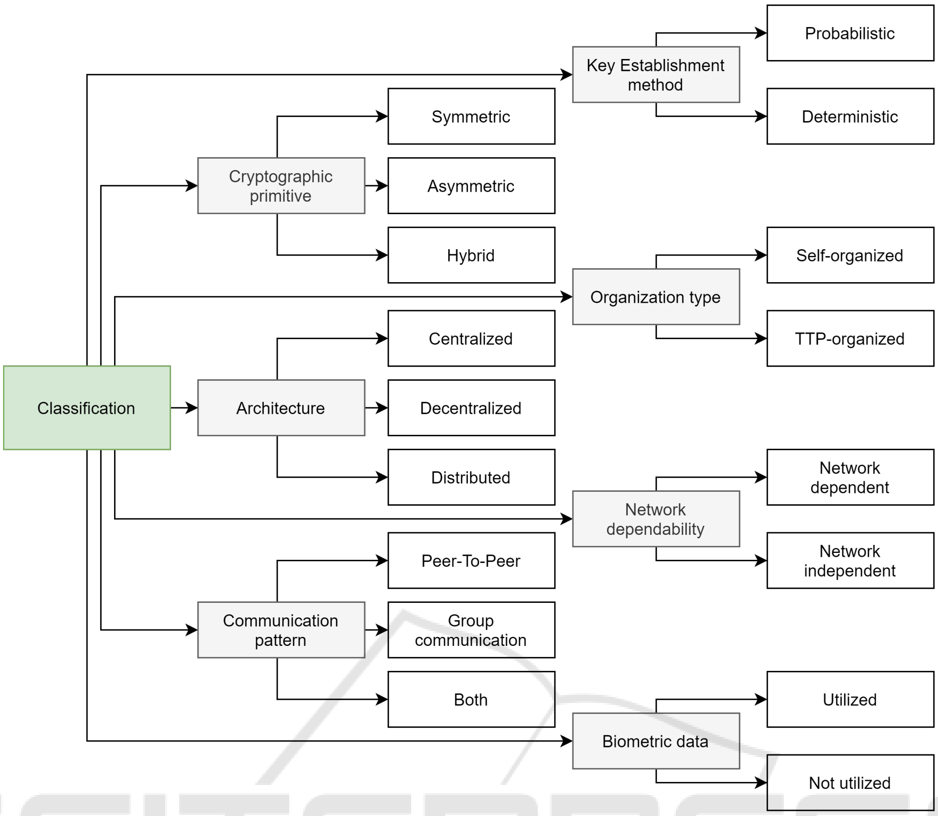

Figure 1: Classification of key management approaches (Jank, 2020).

Asymmetrical approaches perform better in point-to-point communication with multiple individual entities. If a unique symmetric key is used for every connection, $(n-1)$ keys must be stored in a secure way per node or a total of $(n(n-1)/2)$ keys system-wide (Haque et al., 2018). As asymmetric encryption utilizes the public key of the communication partner which can be freely published, in an minimal setup only the own private key needs to be protected. Public keys of communication partners can be verified using a TTP.

Different approaches exist to minimize the symmetric keys needed, like matrix-based (Nafi et al., 2020) or key pool-based approaches (Ahlawat and Dave, 2018). Furthermore, asymmetric protocols can be adapted to fit embedded system constrains e.g. (Liu and Ning, 2008; Raza and Mar Magnusson, 2019).

Asymmetric secret keys used for authentication can be generated directly on embedded devices, although depending on the employed trust model they might still need to be certified by a trusted third party. This ensures that as long as the key is not compromised, strong non-repudiation can be guaranteed. Symmetric keys are always shared between at least two entities, not providing non-repudiation (Barker, 2020). This also impacts key storage, as asymmetric keys only need to be protected on one device, while symmetric keys must be protected on all communication partners.

Another advantage of asymmetric keys is that when generated directly on the device, the private key never needs to be transported to another device (Barker, 2020). Symmetric keys always have to be transmitted which can affect confidentiality and integrity of the key.

3.2 Architecture

In centralized approaches there exists a designated entity exclusively responsible for specific tasks e.g. creation and distribution of keys (Devaraju and Ganapathi, 2010; Sharma and Krishna, 2015; Vasukidevi and Sethukarasi, 2017). Examples for a designated entity are the group leader of a clustered approach, a KDC in a symmetric approach or a CA in an asymmetric approach. When a set of devices need a common key to securely encrypt communication between each other, only one entity is responsible to distribute that group key (Devaraju and Ganapathi, 2010).

Decentralized approaches share the responsibilities between multiple entities. In cluster based approaches for example, there exists a designated device in each sub-group, called LC, which is responsible for handling certain key management tasks (Devaraju and Ganapathi, 2010; Vasukidevi and Sethukarasi, 2017). This device often is less constrained than other members of the cluster. When sticking to the example used before, each group leader would be responsible to distribute the group key to its groups members (Devaraju and Ganapathi, 2010). The resulting system will be decentralized but each group for itself would be centralized.

In distributed approaches group members cooperate to fulfill certain key management objectives. No exclusively responsible entity exists. To stick to the previous example, in distributed approaches devices work together to agree upon a group key (Devaraju and Ganapathi, 2010; Vasukidevi and Sethukarasi, 2017).

Centralized approaches are the most widely used and researched (Guoyu Xu et al., 2012; Menesidou et al., 2017; Harn et al., 2018). Their main advantages are the efficiency in both transmission and computation. Common problems in centralized approaches are that they induce a *single-point of failure* (Liu et al., 2020) and can be a bottleneck in big networks (Devaraju and Ganapathi, 2010). Purely centralized approaches require trust into the entity generating and distributing secret keys (Harn et al., 2018). Centralized approaches do not enforce a mutual TTP (Harn et al., 2018). This can be an advantage if the members of a group want to communicate in a secure way and cannot trust the centralized key provider e.g. in an internet chat (Dowsley et al., 2016). In such an attempt a centralized entity can establish keys for a secure connection between the participants but is not involved in further group key establishment. A property and possible limitation is that the dedicated entity providing a service (e.g. key distribution) can only be used by devices if the service is reachable (Menesidou et al., 2017). This can for example become a problem if an otherwise offline systems relies on a cloud KDC. Centralized approaches are prone to infrastructure failures (Menesidou et al., 2017) and "*may suffer from low availability and poor scalability due to the low reliability and poor connectivity of networks*" (Saravanan et al., 2011).

To reduce the key management overhead of large-scale centralized networks, decentralized approaches can be used (Guoyu Xu et al., 2012). The overhead is shared between the LC (Menesidou et al., 2017). If one controller fails this will not affect other groups / clusters (Liu et al., 2020), reducing the effect of the *single-point of failure* problem (Challal and Seba, 2005). Communication between the groups can lead to delays, as data needs to be transmitted via the LC (Liu et al., 2020). As there is no global group key, multiple encryption and decryption operations are required. This can be a system bottleneck (Guoyu Xu et al., 2012) and increase CPU workload and subsequent energy consumption.

Distributed approaches solve the *single-point of failure* problem (Liu et al., 2020). They improve the reliability of the overall system, reduce the bottlenecks in networks (Devaraju and Ganapathi, 2010) and are more tolerant to infrastructure failure (Menesidou et al., 2017). Challenges exist regarding privacy (Menesidou et al., 2017), storage and communication overhead (Guoyu Xu et al., 2012; Liu et al., 2020) when e.g. establishing group keys. This can affect battery lifetime as well as transmission time.

The classification into centralized, decentralized and distributed is most commonly used in group communication (Rafaeli and Hutchison, 2003; Challal and Seba, 2005; Mapoka, 2013; Sharma and Krishna, 2015; Liu et al., 2020).

3.3 Organization Type

Approaches are self-organized if devices "*do not have to rely on an online centralized TTP to provide key management services during network deployment*" (De Ree et al., 2019). Certificate chaining (Capkun et al., 2003) or self-certification-based (van der Merwe et al., 2005) key management can be taken as examples (De Ree et al., 2019). Those approaches rely on a partially distributed TTP. The advantage is that they can handle offline authorization. Disadvantages are that nodes may act selfishly if TTP nodes leave the group, due to the overhead if being selected as new TTP. Further such approaches rely on physical contact to "*instantiate trust and distribute keys*" (De Ree et al., 2019).

When using TTP organized approaches different options exist in what quantity and for which tasks the third party is needed. In e.g. identity-based encryption systems, such approaches the public key is replaced by one or a set of unique public identification parameters (*IDs*) (Drias et al., 2017). To generate a private key, devices transmit their public *IDs* to a TTP called PKG. To encrypt a message for a specific device *D* the public key of the PKG and the public identification parameters of *D* are needed. The TTP is solely needed for initial key generation. After that the system can work in a decentralized mode. Similar to other centralized approaches the PKG knows all keys which may be a violation of the privacy and confidentiality property (key escrow problem) (Drias et al., 2017). To mitigate this problem the PKG can be taken offline as soon as system setup is finished (Drias et al., 2017). An advantage of identity-based approaches and at the same time and open issue is the revocation of keys. The advantage is that there is no need of CRL. If an entity is no longer in the system or unreachable, its identity can't be used to encrypt messages. The issue lies with compromised private keys. In this case outgoing and incoming messages can be manipulated. A solution is to add timestamps to the public identifiers to shorten the cryptoperiod (Boneh and Franklin, 2001). This increases the times the PKG is needed as private keys have to be issued more frequently. Another method is to add a second TTP called Identity Revocation Server (IRS). Prior to communication each entity requests a revocation status for itself and gets it signed. After that the entities can send and receive encrypted messages. This approach to solve the key revocation problem reintroduces the need for a reachable TTP.

3.4 Communication Pattern

Key management approaches may differ by the type of secure communication needed within the managed systems. Communication may happen in peer-to-peer fashion (Kandi et al., 2020) (also called unicast or point-to-point (ISO, 2018)) with other nodes or TTP, within groups (Murugesan and Saminathan, 2018) (also called multicast (Abirami and Padmavathy, 2017)) or a combination of those. Key management faces different challenges depending on the utilized communication patterns.

Unicast (Point-To-Point) communication needs unique keys between two entities, which is usually called pairwise key (Kandi et al., 2020). Messages can only be read by these two entities. Those should be agreed on each session anew and be only used once. Challenges are in establishing the first trust as well as authentication between nodes. Furthermore, key storage can become an issue when numerous point-to-point connections are needed. Unicast communication suffers from poor scalability and flexibility. Different approaches exist to mitigate this limitations (Kandi et al., 2020).

In group communication a set of entities communicates with each other. The exchanged messages are readable by the entire group. A common key must be known exclusively by current members (Kandi et al., 2020). Challenges are within group key establishment and rekeying if an entity leaves or enters the group. Entities leaving the group should not be able to read future messages (forward secrecy) and new members should not be able to decipher old messages (backward secrecy) (Kandi et al., 2020). Another challenge is communication between groups as separate keys are needed. Furthermore, group joining, group leaving and compromise of a group member add additional complexity.

Existing solutions still have problems with the heterogeneity of embedded systems (Kandi et al., 2020). Additionally, the same parameters are used independently of the submitted data and load is not balanced fairly between powerful and weak nodes (Kandi et al., 2020).

All communication patterns are vulnerable to so-called node capture attacks. The less communication can be deciphered if a node is compromised the more resilient this node is against capture (Zhen Yu and Yong Guan, 2005). To maximize resilience a distinct key is needed for each pair of devices (Kandi et al., 2020). As described in Section 3.2 this would have negative effects for embedded systems.

3.5 Key Establishment Approach

This classification focuses on key establishment methods and divides them into deterministic and probabilistic (Messai and Seba, 2016; Mesmoudi et al., 2019). A probabilistic approach was first proposed by Eschenauer and Gligor in (Eschenauer and Gligor, 2002). In such approaches for each member of a network, a set of keys is chosen, in a random or semi random way (Abu Al-Haija, 2011). The pool of keys can be generated by an external server or directly on the devices (Leshem et al., 2018). After each entity is equipped with a set of keys, commonly called key ring (Abu Al-Haija, 2011), two nodes in the network share the same key with a predefined probability. This probability is depending on the method used. If two nodes share any common keys, they can establish a secure connection between each other.Otherwise they can try to establish a connection by using other devices as proxies. In the worst case two nodes cannot establish a connection between each other.

Conversely in deterministic approaches nodes have a probability of 1 *(sure event)* to share a common key with all nodes (Messai and Seba, 2016). They have less communication overhead as they do not need to route communication via other devices (Mesmoudi et al., 2019).

Probabilistic approaches try to improve efficiency and security by minimizing the keys saved per device. This has a positive effect on storage capacity and improves resilience against node capture attacks (Eschenauer and Gligor, 2002). However Xu et al. argue that these advantages do not have such a great impact as presumed and have to be opposed to the increasing complexity and low network connectivity

(Xu et al., 2007).

If the amount of nodes in a system increases, storage capacity can becomes a bottleneck for constrained devices. Probabilistic approaches can reduce the burden but have an impact on communication performance.

3.6 Biometric Data

Biometric approaches extract various biometric features to create biometric keys (Masdari et al., 2017). Biometric data can be physiological or behavioral. The former are measurements of the human body e.g. fingerprint, iris, retina, hand geometry or face and the latter are measurements based on human actions e.g. signatures, keystrokes or voice (Masdari et al., 2017). As such approaches need physical presence, their use-cases are limited as they need at least one sensor extracting some biometric feature. WBAN are one sector which can use biometric data at no cost as their sensors mostly already extract those.

Biometrically generated keys can not be used for conventional cryptographic schemes as those by design do not tolerate even a single-bit error (Bui and Hatzinakos, 2007). The challenge in such approaches is to generate constant and repeatable keys from variant biometric samples (Chen et al., 2007). A method to circumvent this problem is to divide biometric identifiers into a set of separate elements. When recalculating the key not all measurements have to be correct to be able to successfully retrieve the key. Moreover, biometric data comes with privacy and security concerns (Cavoukian, 2007). Storage and misuse for surveillance, profiling etc. is still an issue.

Advantages are that biometric approaches can generate keys from sensor data. Approaches exist needing live biometric samples for verification and through that prevents dictionary and brute-force attacks (Chen et al., 2007).

If exterior features such as face, fingerprint or retina are used they can be stolen and bypassed by e.g. taking a picture of the victims face or directly exploiting latent fingerprints remaining on a smartphones fingerprint reader (scrap attack) (Kim et al., 2017).

3.7 Network Topology

Systems can be classified into flat and hierarchical (Mesmoudi et al., 2019). In flat systems, all nodes have the same capabilities in the sense of computational power, battery life, storage, memory size, etc. This is different in hierarchical systems. In these systems, some devices can be more constrained then others. This fact can be utilized by key management approaches to increase performance by distributing tasks depending on device capabilities (Albakri et al., 2019). Those approaches are called hierarchical approaches. For instance in cluster based systems it can be of advantage if the LC (see section 3.2) is less constrained. That enables it to take over computationally heavy tasks. This can improve the overall efficiency of the system.

3.8 Network Dependability

To operate efficiently, network dependent approaches rely on the features of the underlying network infrastructure (Mapoka, 2013). Conversely, network independent approaches can be used in different architecture setting (Mapoka, 2013).

4 CONCLUSIONS AND OUTLOOK

In this work we briefly summarized our first steps towards creating a unifying taxnonomy for supporting developers in the IoT context with the selection of appropriate key management approaches.

We provided a first glance at our classification, albeit still omitting important aspects; due to space constraints we were for example unable to detail the factors influencing the actual decision for a specific approach, be they internal (like existing know-how and infrastructure, policies, or strategies), external (like legislation, customers, competitors), or product-driven (like environmental conditions, process needs, project size and of course actual security requirements).

So there is still a lot of work ahead in order to close this existing gap, establishing a generic decision supporting mechanism to enhance security aspects in the Internet of Things and embedded systems, respectively.

ACKNOWLEDGEMENTS

Parts of this work were supported by the Embedded Lab Vienna for IoT & Security, funded by the City of Vienna.

Funded by City of Vienna Economic Affairs, Labour and Statistics ELVIS EMBEDDED LAB VIENNA FOR IOT & SECURITY

REFERENCES

Abirami, E. and Padmavathy, T. (2017). Proficient key management scheme for multicast groups using group key agreement and broadcast encryption. In *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, Chennai, India. IEEE.

Abouhogail, R. A. (2014). Security Assessment for Key Management in Mobile Ad Hoc Networks. *International Journal of Security and Its Applications*, 8(1):169–182.

Abu Al-Haija, Q. (2011). Toward Secure Non-Deterministic Distributed Wireless Sensor Network Using Probabilistic Key Management Approaches Qasem Abu Al-Haija. *Journal of Information Assurance and Security*, 6.

Abusukhon, A., Mohammad, Z., and Al-Thaher, A. (2019). Efficient and Secure Key Exchange Protocol Based on Elliptic Curve and Security Models. In *2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)*, pages 73–78, Amman, Jordan. IEEE.

Ahlawat, P. and Dave, M. (2018). Deployment Based Attack Resistant Key Distribution with Non Overlapping Key Pools in WSN. *Wireless Personal Communications*, 99(4):1541–1568.

Albakri, A., Harn, L., and Song, S. (2019). Hierarchical Key Management Scheme with Probabilistic Security in a Wireless Sensor Network (WSN). *Security and Communication Networks*, 2019:1–11.

Alcaraz, C., Lopez, J., Roman, R., and Chen, H.-H. (2012). Selecting key management schemes for WSN applications. *Computers & Security*, 31(8):956–966.

Alohali, B., Kifayat, K., Shi, Q., and Hurst, W. (2015). A Survey on Cryptography Key Management Schemes for Smart Grid. *Journal of Computer Sciences and Applications*, 3(3A):27–39.

Balasubramanian, A., Mishra, S., and Sridhar, R. (2005). Analysis of a hybrid key management solution for ad hoc networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pages 2082–2087, New Orleans, LA, USA. IEEE.

Barker, E. (2020). Recommendation for Key Management: Part 1 - General. Technical report.

Barker, E. and Barker, W. C. (2019). Recommendation for key management: Part 2 – best practices for key management organizations. Technical Report NIST SP 800-57pt2r1, Gaithersburg, MD.

Barr Group (2018). *2018 Embedded Systems Safety & Security Survey Report*.

Barskar, R. and Chawla, M. (2016). A Survey on Efficient Group Key Management Schemes in Wireless Networks. *Indian Journal of Science and Technology*, 9(14).

Boneh, D. and Franklin, M. (2001). Identity-Based Encryption from the Weil Pairing. In Goos, G., Hartmanis, J., van Leeuwen, J., and Kilian, J., editors, *Advances in Cryptology — CRYPTO 2001*, volume 2139, pages 213–229. Springer Berlin Heidelberg, Berlin, Heidelberg.

Bui, F. M. and Hatzinakos, D. (2007). Biometric Methods for Secure Communications in Body Sensor Networks: Resource-Efficient Key Management and Signal-Level Data Scrambling. *EURASIP Journal on Advances in Signal Processing*, 2008(1).

Capkun, S., Buttyan, L., and Hubaux, J. (2003). Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing*, 2(1):52–64.

Cavoukian, A. (2007). Biometric Encryption : A Positive-Sum Technology that Achieves Strong Authentication , Security AND Privacy.

Challal, Y. and Seba, H. (2005). Group Key Management Protocols: A Novel Taxonomy. *Information Technology - IT*, 2.

Checkpoint (2018). Faxploit HP Printer Fax Exploit.

Chen, H., Sun, H., and Lam, K.-Y. (2007). Key Management Using Biometrics. In *The First International Symposium on Data, Privacy, and E-Commerce (ISDPE 2007)*, pages 321–326, Chengdu, China. IEEE.

Dawson, R., Boyd, C., Dawson, E., and Nieto, J. (2006). SKMA: a key management architecture for SCADA systems. In *Conferences in Research and Practice in Information Technology Series*, volume 54, pages 183–192.

De Ree, M., Mantas, G., Radwan, A., Mumtaz, S., Rodriguez, J., and Otung, I. E. (2019). Key Management for Beyond 5G Mobile Small Cells: A Survey. *IEEE Access*, 7:59200–59236.

Devaraju, S. and Ganapathi, P. (2010). Dynamic Clustering for QoS based Secure Multicast Key Distribution in Mobile Ad Hoc Networks. *IJCSI International Journal of Computer Science*, 7.

Dowsley, R., Gabel, M., Hubsch, G., Schiefer, G., and Schwichtenberg, A. (2016). A Distributed Key Management Approach. In *2016 IEEE International Conference on Cloud Computing Technology and Science (CloudCom)*, pages 509–514, Luxembourg, Luxembourg. IEEE.

Drias, Z., Serhrouchni, A., and Vogel, O. (2017). Identity-based cryptography (IBC) based key management system (KMS) for industrial control systems (ICS). In *2017 1st Cyber Security in Networking Conference (CSNet)*, Rio de Janeiro. IEEE.

Eschenauer, L. and Gligor, V. D. (2002). A key-management scheme for distributed sensor networks. In *Proceedings of the 9th ACM conference on Computer and communications security - CCS '02*, page 41, Washington, DC, USA. ACM Press.

European Union and Agency for Network and Information Security (2017). *Baseline security recommendations for IoT in the context of critical information infrastructures.*

European Union Agency for Cybersecurity (2019). *Good practices for security of IoT: secure software development lifecycle.*

Grand View Research (2014). Embedded System Market Report Embedded System Market Analysis By Product (Hardware, Software), By Application (Automotive, Telecommunication, Healthcare, Industrial, Consumer Electronics, Military & Aerospace) And Segment Forecasts To 2020. Technical report.

Guoyu Xu, Xingyuan Chen, and Xuehui Du (2012). Chinese Remainder Theorem based DTN group key management. In *2012 IEEE 14th International Conference on Communication Technology*, pages 779–783, Chengdu, China. IEEE.

Haque, M. E., Zobaed, S., Islam, M. U., and Areef, F. M. (2018). Performance Analysis of Cryptographic Algorithms for Selecting Better Utilization on Resource Constraint Devices. In *2018 21st International Conference of Computer and Information Technology (ICCIT)*, pages 1–6, Dhaka, Bangladesh. IEEE.

Harn, L., Hsu, C.-F., and Li, B. (2018). Centralized Group Key Establishment Protocol without a Mutually Trusted Third Party. *Mobile Networks and Applications*, 23(5):1132–1140.

Huang, X. and Chen, R. (2018). A Survey of Key Management Service in Cloud. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pages 916–919, Beijing, China. IEEE.

Humayed, A., Lin, J., Li, F., and Luo, B. (2017). Cyber-Physical Systems Security—A Survey. *IEEE Internet of Things Journal*, 4(6):1802–1831.

ISO (2018). Information technology - Security techniques - Key management - Part 2: Mechanisms using symmetric techniques. Standard ISO/IEC 11770-2:2018, Geneva, CH.

Jank, G. (2020). Secure key management in embedded systems. Master's thesis, FH Campus Wien - University of Applied Sciences.

Jansma, N. and Arrendondo, B. (2004). *Performance Comparison of Elliptic Curve and RSA Digital Signatures*.

Kandi, M. A., Lakhlef, H., Bouabdallah, A., and Challal, Y. (2020). A versatile Key Management protocol for secure Group and Device-to-Device Communication in the Internet of Things. *Journal of Network and Computer Applications*, 150.

Kim, S., Lee, H., and Kwon, T. (2017). POSTER: Rethinking Fingerprint Identification on Smartphones. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 2515–2517, Dallas Texas USA. ACM.

Kolias, C., Kambourakis, G., Stavrou, A., and Voas, J. (2017). DDoS in the IoT: Mirai and Other Botnets. *Computer*, 50(7).

Lavanya, S. and Usha, D. (2017). A Survey on Key Management in Internet of Things. *SSRN Electronic Journal*.

Leshem, G., David, E., and Domb, M. (2018). Probability Based Keys Sharing for IOT Security. In *2018 IEEE International Conference on the Science of Electrical Engineering in Israel (ICSEE)*, pages 1–5, Eilat, Israel. IEEE.

Liu, A. and Ning, P. (2008). TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. In *2008 International Conference on Information Processing in Sensor Networks (ipsn 2008)*, pages 245–256, St. Louis, MO, USA. IEEE.

Liu, J., Tong, X., Wang, Z., Zhang, M., and Ma, J. (2020). A Centralized Key Management Scheme Based on McEliece PKC for Space Network. *IEEE Access*, 8:42708–42719.

Malik, M., Dutta, M., and Granjal, J. (2019). A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things. *IEEE Access*, 7:27443–27464.

Manikandan, G. and Sakthi, U. (2018). A Comprehensive Survey on Various key Management Schemes in WSN. In *2018 2nd International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), 2018 2nd International Conference on*, pages 378–383, Palladam, India. IEEE.

Mapoka, T. T. (2013). Group Key Management Protocols for Secure Mobile Multicast Communication: A Comprehensive Survey. *International Journal of Computer Applications*, 84(12):28–38.

Masdari, M., Ahmadzadeh, S., and Bidaki, M. (2017). Key management in wireless Body Area Network: Challenges and issues. *Journal of Network and Computer Applications*, 91:36–51.

Menesidou, S. A., Katos, V., and Kambourakis, G. (2017). Cryptographic Key Management in Delay Tolerant Networks: A Survey. *Future Internet*, 9(3):26.

Menezes, A. J., Van Oorschot, P. C., and Vanstone, S. A. (1997). *Handbook of applied cryptography*. CRC Press series on discrete mathematics and its applications. CRC Press, Boca Raton.

Mesmoudi, S., Benadda, B., and Mesmoudi, A. (2019). SKWN: Smart and dynamic key management scheme for wireless sensor networks. *International Journal of Communication Systems*, 32(7).

Messai, M.-L. and Seba, H. (2016). A survey of key management schemes in multi-phase wireless sensor networks. *Computer Networks*, 105:60–74.

Mirjalili, S. H. and Lenstra, A. K. (2008). Security Observance throughout the Life-Cycle of Embedded Systems. In *ESA*.

M.K, J., Abhiraj, K. S., Bolloju, S., Brukbacher, S., Carullo, G., Dhungel, R., Donahoe, T., Duddilla, R., Duren, D. V., Falletta, L., Yeoh, J., Santos, J., Ferrari, L., Figuigui, A., Futagi, M., Grimes, M., Guzman, A., Hendrickson, H., Khemissa, S., Lanois, P., Kannimoola, J. M., Naik, S., Perera, C., Rajapaksha, S., Rupasinghe, L., Sharma, A., Szewczul, M., Tatipamula, S., Thikkavarapu, S., and White, K. (2016). Future-proofing the Connected World: 13 Steps to Developing Secure IoT Products.

Murugesan, A. and Saminathan, B. (2018). Key management for secured group communication. *International Journal of Pure and Applied Mathematics*, 118:33–37.

Nafi, M., Bouzefrane, S., and Omar, M. (2020). Matrix-based key management scheme for IoT networks. *Ad Hoc Networks*, 97.

Nie, S., Liu, L., and Du, Y. (2017). FREE-FALL: HACKING TESLA FROM WIRELESS TO CAN BUS. Keen Security Lab of Tencent.

Obermaier, J., Hauschild, F., Hiller, M., and Sigl, G. (2018). An embedded key management system for PUF-based security enclosures. In *2018 7th Mediterranean Conference on Embedded Computing (MECO)*, pages 1–6, Budva, Montenegro. IEEE.

Payment Security Support Group (2018). European Payments Council: Guidelines on Cryptographic Algorithms Usage and Key Management - Version 8.0. Technical Report EPC342-08.

Rafaeli, S. and Hutchison, D. (2003). A survey of key management for secure group communication. *ACM Computing Surveys*, 35(3):309–329.

Rajendran, G. and Nivash, R. (2019). Security in the Embedded System: Attacks and Countermeasures. *SSRN Electronic Journal*.

Raza, S. and Mar Magnusson, R. (2019). TinyIKE: Lightweight IKEv2 for Internet of Things. *IEEE Internet of Things Journal*, 6(1):856–866.

Sadeghi, A.-R., Wachsmann, C., and Waidner, M. (2015). Security and privacy challenges in industrial internet of things. In *Proceedings of the 52nd Annual Design Automation Conference on - DAC '15*, San Francisco, California. ACM Press.

Saravanan, D., Rajalakshmi, D., and Maheswari, D. (2011). DYCRASEN: A Dynamic Cryptographic Asymmetric Key Management for Sensor Network using Hash Function. *International Journal of Computer Applications*, 18(8):1–3.

Sharma, S. and Krishna, C. R. (2015). An Efficient Distributed Group Key Management Using Hierarchical Approach with Elliptic Curve Cryptography. In *2015 IEEE International Conference on Computational Intelligence & Communication Technology*, pages 687–693, Ghaziabad, India. IEEE.

Vai, M., Nahill, B., Kramer, J., Geis, M., Utin, D., Whelihan, D., and Khazan, R. (2015). Secure architecture for embedded systems. In *2015 IEEE High Performance Extreme Computing Conference (HPEC)*, pages 1–5, Waltham, MA, USA. IEEE.

van der Merwe, J., Dawoud, D., and McDonald, S. (2005). Fully self-organized peer-to-peer key management for mobile ad hoc networks. In *Proceedings of the 4th ACM workshop on Wireless security - WiSe '05*, page 21, Cologne, Germany. ACM Press.

Vasukidevi, G. and Sethukarasi, D. R. (2017). A Survey on Security and Key Management in VANET.

Wadhwani, P. and Yadav, S. (2020). Embedded systems market size by component (hardware, [asic & assp, microcontroller, microprocessor, power management integrated circuit (pmic), field programmable gate array (fpga), digital signal processor (dsp), memory], software (os, middleware)], by function (standalone system, real-time system, network system, mobile system), by application (automotive, consumer electronics, manufacturing, retail, media & entertainment, military & defense, telecom), industry analysis report, regional outlook, application potential, competitive market share & forecast, 2020 – 2026. Technical report.

Wan, J., Lopez, A. B., and Al Faruque, M. A. (2016). Exploiting Wireless Channel Randomness to Generate Keys for Automotive Cyber-Physical System Security. In *Proceedings of the 7th International Conference on Cyber-Physical Systems*, ICCPS '16. IEEE Press.

Whelihan, D., Vai, M., Utin, D., Khazan, R., Gettings, K., Anderson, T., Godfrey, A., Govotski, R., Yeager, M., Chetwynd, B., Nahill, B., and Koziel, E. (2016). SHAMROCK: A Synthesizable High Assurance Cryptography and Key management coprocessor. In *MILCOM 2016 - 2016 IEEE Military Communications Conference*, pages 55–60, Baltimore, MD, USA. IEEE.

Xu, D., Huang, J., Dwoskin, J., Chiang, M., and Lee, R. (2007). Re-examining Probabilistic Versus Deterministic Key Management. In *2007 IEEE International Symposium on Information Theory*, pages 2586–2590. IEEE.

Zhen Yu and Yong Guan (2005). A robust group-based key management scheme for wireless sensor networks. In *IEEE Wireless Communications and Networking Conference, 2005*, volume 4, pages 1915–1920, New Orleans, LA, USA. IEEE.