Secure Key Management in Embedded Systems: A First Proposal

Gorazd Jank, Silvia Schmidt and Manuel Koschuch

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 (Koliias 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 (Koliias 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...
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 start-
ing 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 se-
curity 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 de-
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 secu-
rrity requirements and define processes on how to en-
force them. During implementation, different secu-
rity mechanisms will be needed to safeguard differ-
ent parts of the solution. Those often need secrets,
which in most cases will be realized through crypto-
graphic keys. The assumption that a single key will
remain secure for the entire lifetime of the embed-
ded system typically is not sufficient (Grand View Re-
search, 2014). To account for revocation of compro-
mised keys and generation of new ones, key manage-
ment 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 man-
agement. These differ by the usage of special hard-
ware 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 man-
agement solution. The remainder of this paper is now
structured as follows: Section 2 gives a comprehen-
sive overview of existing works dealing with the prob-
lem of key management in constrained devices. From
that, we present our contributions in form of a tax-
onomy 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 de-
defined in 1997’s Handbook of Applied Cryptography
by Menezes et al. (Menezes et al., 1997). This in-
cludes 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 plan-
ing of key management. They start by choosing a fit-
ting key management architecture based on the avail-
able 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, in-
formation about key generation and/or distribution,
key storage, access control, accounting and audit-
ing, 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 de-
veloped. This is “a set of rules that are established to
describe the goals, responsibilities, and overall require-
ments 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 organiza-
tion and a CKMS Practice Statement is developed.
It describes the establishment of a trust root for the
CKMS and a specification “how key management proce-
dures and techniques are used to enforce the CKMS
Security Policy and to conform with the Key Manage-
ment Specification” (Barker and Barker, 2019). This
planning procedure does not consider possible limita-
tions which embedded devices can have.

Alcaraz et al. (Alcaraz et al., 2012) present an ap-
proach on how to map WSN requirements to key man-
gagement protocol properties. They defined a set of at-
tributes 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 (Alohal 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 ap-
proaches are separated into symmetric and asymmet-
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 that for 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 connectiv-

ity, 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 Networks (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 STTX 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 Sathi, 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. Fur-
ther 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).
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 a 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 ap-
proaches 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 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 introduces 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 Padmatvathy, 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 these 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.
If the amount of nodes in a system increases, storage capacity can become 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 cannot 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 victim's face or directly exploiting latent fingerprints remaining on a smartphone's 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 taxonomy 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.
REFERENCES


Grand View Research (2014). Embedded System Market Report Embedded System Market Analysis By Product (Hardware, Software), By Application (Automotive, Telecommunication, Healthcare, Industrial, Con

89
Security enclosures. In 2018 7th Mediterranean Conference on Embedded Computing (MECO), pages 1–6, Budva, Montenegro. IEEE.


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.


