Comparison of Energy-Efficient Key Management Protocols for Wireless Sensor Networks

Laurin Doerr
University of West Bohemia, Pilsen, Czech Republic
Deggendorf Institute of Technology,
Deggendorf, Germany
laurind@kiv.zcu.cz, laurin.doerr@th-deg.de

Michael Heigl
University of West Bohemia, Pilsen, Czech Republic
Deggendorf Institute of Technology,
Deggendorf, Germany
heigl@kiv.zcu.cz, michael.heigl@th-deg.de

Dalibor Fiala
University of West Bohemia,
Pilsen, Czech Republic
dalfia@kiv.zcu.cz

Martin Schramm
Deggendorf Institute of Technology,
Deggendorf, Germany
martin.schramm@th-deg.de

ABSTRACT

A Wireless Sensor Network (WSN) contains small sensor nodes which monitor physical or environmental conditions. WSN is an important technology for digitalization of industrial periphery and is often used in environments which are not hardened against security impacts. These networks are easy to attack due to the open communication medium and low computing resources of the applied devices. Establishing security mechanisms is difficult while taking into account low energy consumption. Low cost sensors with limited resources make the implementation of cryptographic algorithms even more challenging. For WSNs cryptographic functions are needed without high impact on energy consumption and latency. Therefore, security in WSNs is a challenging field of research. This paper compares lightweight energy-efficient key exchange protocols which are suitable for WSN. The protocols were also implemented in WSN-capable Texas Instrument boards and the energy consumption was measured during the key exchange. This paper shows that schemes have to be chosen depending on the specific network requirements and that the usage of asymmetric cryptography does not always result in a high energy consumption.

CCS Concepts

- **Security and privacy→ Mobile and wireless security**
- **Security and privacy→ Key management**
- **Security and privacy→ Security protocols**

Keywords

Wireless sensor networks; Key management; Energy consumption

1. INTRODUCTION

Nowadays, an increasing number of devices are used to monitor our environment and connect embedded objects together. They often use the Wireless Sensor Network (WSN) technology to ensure connectivity between nodes at the lower level of network architecture. WSNs are wireless networks composed of distributed devices, which are often simple sensors. The sensors are used to monitor physical or environmental conditions, such as vibrations, temperature or motion [5]. Every sensor device is powered by a battery and equipped with a radio transceiver. Due to the flexibility of deployment simplicity, WSNs are used today in various application areas that also include industrial environments. However, there are some security issues with WSN. A wireless channel is open for every user. Anyone can monitor or participate in communications. Only a radio transceiver configured at the same frequency band is needed. This provides an easy way for intruders to break into WSNs. In such a context, sensitive applications require security mechanisms to achieve the three main security goals: confidentiality, integrity and availability. For most applications of WSNs, a high level of security is required [12]. Therefore, it is important to design secure communication mechanisms between all devices of the network. Because the nodes often have limited computing resources and a limited battery lifetime, an efficient key exchange algorithm has to be used. In networks with high bandwidth available, common algorithms such as RSA-2048 are utilized. These algorithms often need a lot of computing power and therefore they are not usable in WSNs. Symmetric algorithms are typically characterized by a low computing time and therefore result in a low energy consumption. However, symmetric algorithms depend on an initial secure key exchange. This has to be done in a secure environment and is also a challenge for adding new nodes to the WSN. The application of asymmetric algorithms for the initialization avoid these challenges. After the initialization, symmetric ciphers can be used for encrypting the data exchange. In this paper various lightweight key management schemes are compared. The rest of the paper is organized as follows: Chapter 2 presents four candidate protocols for WSN. In chapter 3 the energy consumption measurements of their key exchange implementations are discussed. Chapter 4 contains a conclusion with remarks to the cryptographic methods and future work of this ongoing research.

2. EFFICIENT KEY MANAGEMENT PROTOCOL

Key management protocols include various parts: Key generation, key exchange, key agreement and key revocation. Key generation is used for the creation of new keys. Key exchange is needed for transferring keys between the associated devices. With key agreement the devices can create derived keys of the original ones

IECC ' 19, July 7–9, 2019, Okinawa, Japan
© 2019 Association for Computing Machinery.
ACM ISBN 978-1-4503-7177-3/19/07…$15.00
DOI: https://doi.org/10.1145/3343147.3343156

and key revocation is needed for invalidation of keys which maybe compromised. In the following sections four key management protocols are described, which can be used as a fundamental part for securing communication in WSNs. There are a lot more protocols in the research field, but these four represent different categories of cryptographic protocols: lightweight symmetric, scalable symmetric, nonstandard and asymmetric cryptography. The Lightweight Authentication Scheme (LAS) was used as an example for a very light and energy efficient protocol. Improved Key Distribution Mechanism (IKDM) can be used in large WSN networks. The IKDM scales very good with the number of nodes in the WSN and is therefore used as an example for a scalable symmetric scheme. The Modified Secured Query Processing Scheme (MSQPS) uses not standardized cryptographic protocols. MSQPS also scales like IKDM with the numbers of nodes and can also be used in cluster-based WSN. As an asymmetric scheme, Key Revocation and Renewal Protocol (KRRP) was selected. For key exchange and authentication of devices, elliptic curves are utilized. For comparing the protocols, five security goals can be used in the context of WSNs [11]: confidentiality, integrity, availability, authenticity and data freshness.

2.1 Lightweight Authentication Scheme

LAS for WSN is proposed in [4] and the scheme was developed for being very light and energy efficient. It provides a key management and an authentication protocol. It uses only symmetric encryption algorithms. This is good for energy consumption and needed processing time. It has no specific requirements on the network topology. The memory requirement is very small. LAS defines three phases: The key pre-distribution phase, the network initialization phase and the authentication phase. The key pre-distribution phase is during network installation in which a master key will be distributed. The network initialization phase is used to generate pairwise keys of each node neighbor. The keys are generated through a random number and the master key. The nodes then calculate authentication keys based on a hash of the master key. All nodes can now forget the master key. The authentication protocol is used for new nodes which join the network. It is efficient by design and needs only three messages for authentication. The authentication protocol is mainly created for static scenarios in which the expected rate of new nodes and authentications is low. The authentication protocol can also not authenticate a specific device based on any signatures. The freshness of the key material data is guaranteed through a nonce mechanism during key exchange. LAS protects from physical attacks and Denial of Service (DoS) attacks. Therefore, LAS provides mechanisms for the security goals availability and integrity. However, using a master key at installation and initialization is a potential risk. If an intruder can claim a device before the network initialization phase, he gets access to the network at any time. This implies that the first two phases have to be run in a secure environment. This can often not be achieved in an industrial environment. Another disadvantage of LAS is that the master key has to be stored for new nodes on a secure place. Otherwise no new node can be authenticated. This means confidentiality depends on the master key. Also no key revocation mechanism is implemented such that a compromised node cannot be isolated from the network even when the node is detected.

2.2 Improved Key Distribution Mechanism for Large-Scale Networks

IKDM for large-scale networks is presented in [3]. The mechanism can maintain the network connectivity even when nodes are compromised. It uses symmetric keys and is specially designed for cluster WSN. The scheme is based on a three-tier hierarchical WSN model and has three phases: the key pre-distribution phase, the inter-cluster pairwise establishment phase and the inter-cluster pairwise key establishment phase. In the key pre-distribution phase, different secret information is loaded to the sensors. During the second phase, two bivariate symmetric polynomials are used to establish pairwise keys between cluster heads and their sensors. In the last phase, the inter-cluster pairwise key establishment phase, each cluster head establishes a pairwise key with other cluster heads. Compared to LAS, IKDM can achieve better network stability when an intruder attacks a node since he can only propagate within one cluster. However, when the node is captured, the shared keys can be used for communication with neighbors because there is no revocation process. This problem is addressed by [2] in which a distributed collaborative key revocation mechanism is proposed. Nodes collaborate in each cluster to identify a malicious node. Then the base station sends a broadcast message containing a list of keys to revoke. Compared to LAS, IKDM also needs more computing time because of more complex functionality. IKDM provides like LAS availability, integrity and freshness during the key exchange. However, the freshness is just a nonce value but it has the same problems with the pre-distributed keys. So confidentiality depends also on a secure deployment of these keys. The attack surface is smaller because of the used polynomials keys.

2.3 Modified Secured Query Processing Scheme

MSQPS provides security mechanisms in a query processing environment and is presented in [6]. It is like the IKDM designed for cluster WSN and also scales with the number of nodes. MSQPS is based on SQPS which is presented in [7]. In MSQPS the Base Station (BS) performs the registration of the Cluster Heads (CH). The CHs then perform the registration of the nodes. The key exchange has two main phases: the query phase and the query response phase. The query phase has two sub-phases: the query phase *BS to CH* and query phase *CH to Node*. The query phase *CH to Node* again has three sub-phases: the registration phase *CH to Node*, the registration response phase *Node to CH* and query forwarding phase *CH to Node*. The MSQPS does not use pre-deployed keys on any node. This is an advantage compared to LAS and IKDM. But for network initialization it has also to be ensured that the devices are not compromised. If a new node has to be registered to the network at a later point of time, the BS has to be informed first. Key revocation is not considered by the scheme. The solution is lightweight and has also a low energy consumption based on its short computing time. Only for network initialization a higher computing time is necessary. The energy consumption should be on a level with IKDM. MSQPS is robust against replay attacks and DoS attacks. It achieves the basic security goals for confidentiality, integrity and availability. Also the freshness of the data during the key exchange phases is guaranteed through timestamping. However, the timestamp is not secured against manipulation. A disadvantage of MSQPS is, that it doesn't use standardized encryption techniques. MSQPS uses bit-shifting functions for encrypting data. The used techniques are not hardened against sniffing attacks during the key exchange. So it cannot automatically provide integrity and confidentiality. The key distribution phase has to be secured against sniffing. Authenticity cannot be achieved due to the missing individual signature of the nodes.

2.4 Key Revocation and Renewal Protocol

The scheme KRRP is proposed in [10] and uses centralized protocols for revoking and renewing keys. It uses symmetric and asymmetric cryptography. To reduce the high computing time for asymmetric keys, Elliptic Curve Cryptography (ECC) is used. As curve for ECC, secp160r1 is recommended. The KRRP can be used in any type of WSN and is not restricted to a special topology. It also supports multi-hop authentication. Six different protocols are defined by KRRP: the join protocol, the revocation protocol, the renewing symmetric key protocol, the renewing asymmetric key protocol, the renewing network key protocol and the multi-hop shared key protocol. Every protocol has different versions for several use cases. The asymmetric keys are used to establish a secure initial key exchange. Every node computes its own private and public key which also allows signing messages. The symmetric keys are used for the encrypted transmission of data. There are two types of keys. The first, computed by the renew symmetric key protocol, is used for the transmission of data between two nodes. The second, computed by the renew network key, is used for end to end data encryption. This increases the confidentiality of the communication if an intruder captures a node. He can only read the data which is specially send to the node. For network initialization, a pre-shared network key can be used to make the first asymmetric key exchange more trustworthy. The KRRP scheme can provide all necessary goals for security in WSN. The freshness is based on a nonce. The KRRP provides a good security basis due to its strong key management protocols. The disadvantage of KRRP is the computing time. It is efficiently designed but compared to the other protocols it needs more energy due to the higher key length compared to symmetric ciphers and the calculation effort [9]. This makes it hard for the application in ultra-low power nodes. In addition, more memory is needed since apart from the public keys also the symmetric keys have to be stored on every data receiving node.

3. ENERGY CONSUMPTION FOR KEY MANAGEMENT

For WSN, low energy consumption is very important as it often uses batteries with a low energy capacity. In the following, the four presented key management protocols were tested on an ARM Cortex-M3 to compare the needed energy and time of protocol execution. For the energy measurement, self-developed implementations of all protocols have been used.

3.1 Testbed

Two CC1350-LaunchPad boards from Texas Instruments (TI) were used for the evaluation. These boards contain the CC1350 microcontroller from TI with a Cortex-M3 processor, which is clocked at 48 MHz. There is also a combined sub-1 GHz radio which also supports Bluetooth Low Energy. For the tests, only the sub-1 GHz part was used and operated at a carrier frequency of 868 MHz. The used programming environment was Code Composer Studio from TI version 8.1.0. The programs are based on the *EasyLinkEchoRx_CC1350_LAUNCHXL_nortos_gcc* and the *EasyLinkEchoTx_CC1350_LAUNCHXL_nortos_gcc* project. These projects send a ping-data-packet between two CC1350-LaunchPads back and forth and serve as a functional example. As libraries for the implementation of cryptographic functions two open source software implementations targeted for ARM embedded processors have been used. For all symmetric functions the *cifra*[1] and for ECC *uECC*[2] was used except for MSQPS which does not use standardized cryptographic functions. Random numbers are generated through the standard C-function *rand()* which are pseudorandom numbers and are not necessarily cryptographically secure. For the energy measurement, an external voltage source with 3.3V was provided by a laboratory power supply. The programmer integrated on the CC1350-LaunchPad has been electrically disconnected to prevent influences on the measurement results. In addition, the LEDs on the boards were not used. The power consumption was recorded via a 10 Ω shunt resistor with an oscilloscope. All programs were executed in a loop. The measuring ranges are highlighted by two vertical lines in the following figures. After a complete key exchange there is a short waiting interval to get an indication in the current curve for the end of one interval.

First, the power consumption of the Ping example was measured to get the energy consumption. of a simple data transmission. It should be noted that 30 bytes are sent, so the energy consumption for sending the ping is high, since each byte extends the send peak. However, the transmission power consumption rises not linear to the number of transmitted bytes.

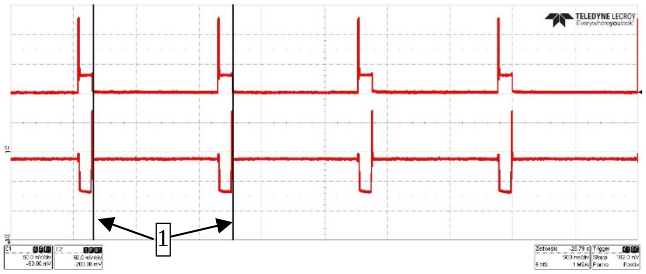

Figure 1. Complete power consumption of Ping

Figure 1 shows the entire interval of the Ping transmission between sender and receiver. The interval is highlighted by two vertical lines (1). There is no waiting time interval because the ping example ends directly after the answer of the second node. The upper curve shows the node that initiates the transfer. The bottom curve represents the current flow of the listening node. Before the transmission begins there is long interval in which the initiator prepares the transmission. The length of the interval also depends on the waiting time of the TI-protocol between every wireless communication. In this phase the listening node is in an active listen state.

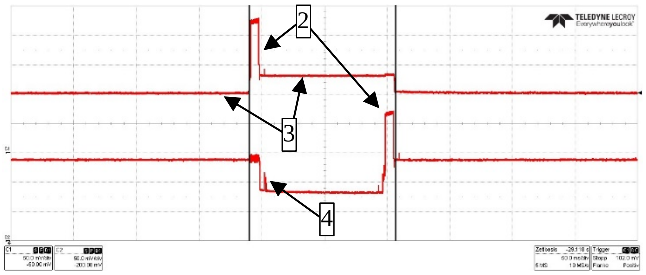

Figure 2. Enlarged power consumption segment of Ping

Figure 2 shows the enlarged power consumption segment of the data transmission without the preparation of the first transmission. The vertical lines indicate the start of the data transmission of the initiator node and the end of the data transmission of the second node. The peaks (2) in the curves are caused by sending data. The

[1] https://github.com/ctz/cifra

[2] https://github.com/kmackay/micro-ecc/

communication initiator begins by transmitting the data and goes into the listening mode after completion. The listening (3) mode needs more power than preparation a transmission because the receiver in the microcontroller is active. The second node copies the data (4) from the receiver into the internal memory after successfully receiving the ping. This is shown by the small peak in the bottom line after the transmission ends. Now the second node prepares the transmission back to the initiator node. The total energy consumption of a complete ping transfer is 86.28 mJ. The Ping example needs 1.12 s for transmitting the data from on node to the other and transmitting it back to the transmission initiator node.

3.2 Lightweight Authentication Scheme

For LAS, PBKDF2 was used to derive a key from the pre-distributed key since it is the most widely used standard for key derivation. However, the necessary computational effort is high which is also noticeable in the energy consumption. The used hash algorithm was SHA-256. 300 rounds were chosen, although 1000 rounds are recommended. In the case of the CC1350, however, a non-detachable stack overflow occurs on over 300 round. An energy consumption of 141.67 mJ was measured. This value is high for the simplicity of LAS. However, PBKDF2 combined with SHA-256 provide a great deal of security for the non-back calculation of the initial pre-shared key. The measurement results are shown in Figure 3 and Figure 4.

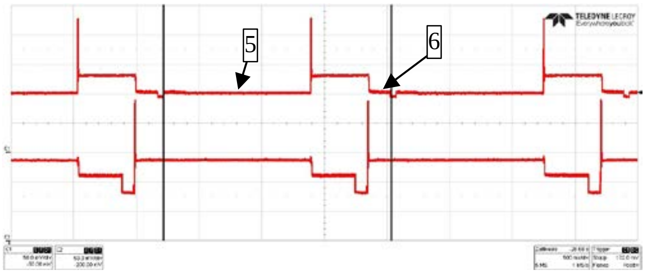

Figure 3. Complete power consumption of LAS

Figure 3 shows the power consumption for the key exchange. The preparation phase (5) before the first transmission takes more time than in the Ping example. This is due to the computation effort for the PBKDF2. After the answer of the second node (second peak), there is an additional time interval (6) before the waiting interval begins. In this interval the derived key is calculated. The complete key exchange needs 1.83 s. Figure 4 shows the two send commands in detail. The small peak (7) in the current curve after the transmission of the initiator node is wider and hard to distinguished from the receiving interval due to more bytes being copied from the receiver into the memory of the node. The second node than also compute the PBKDF2. This needs more power (8) than the preparation of the transmission to the initiator node.

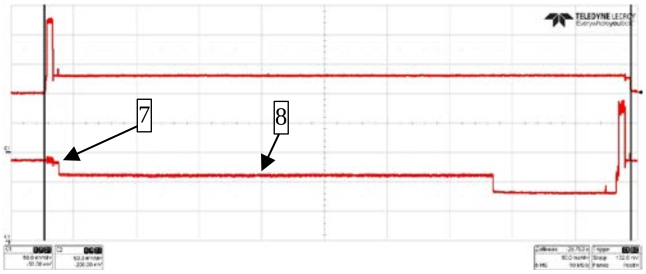

Figure 4. Enlarged power consumption segment of LAS

3.3 Improved Key Distribution Mechanism for Large-Scale Networks

IKDM uses AES-128 encryption in CBC-mode, which from today's point of view is still sufficiently secure. The key exchange between the cluster head and the base station is similar. For the key exchange, data must be exchanged three times between the cluster head and the sensor node. The measured energy consumption is 256.88 mJ.

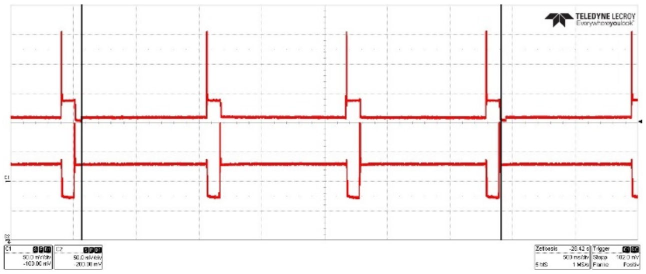

Figure 5. Complete power consumption of IKDM

Figure 5 shows the entire transmission of IKDM. The three transmissions are clearly visible. The use of AES-128 in combination with the assembler-optimized *cifra* library for AES is very efficient. The preparation time of the initiator node is nearly the same as in the Ping example. The sequence ends directly after the last transmission of the second node.

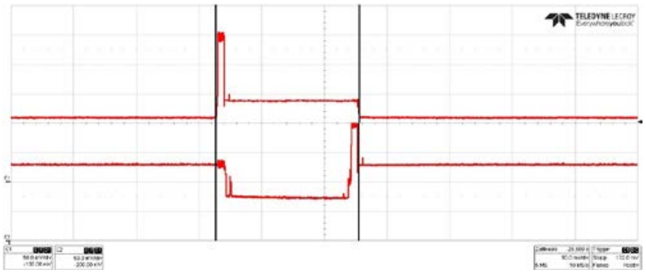

Figure 6. Enlarged power consumption segment of IKDM

Figure 6 shows in detail the data transfer. This excerpt is largely the same in all three transmissions. The interval between the initiator transmission and the receiver node transmission is also nearly the same as in the Ping example and the smaller key size compared to LAS transmission results in nearly the same peak for transferring the data from the receiver to the memory.

3.4 Modified Secured Query Processing Scheme

For MSQPS, the necessary bit-shift functions are not assembler-optimized and neither *cifra* nor *uECC* was used. Therefore, the MSQPS protocol has the highest measured energy consumption with 261.42 mJ. As with the IKDM, three data transfers are necessary.

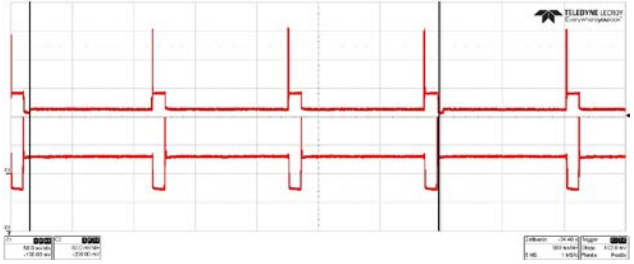

Figure 7. Complete power consumption of MSQPS

Figure 7 again shows the entire measured range. The actual data transmission is similar to the IKDM constant in all three sections. The computation of the data for the key exchange takes also nearly the same time as IKDM.

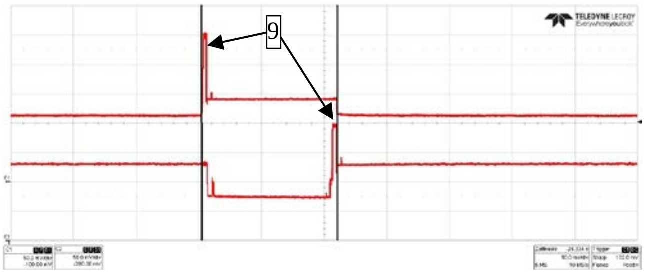

Figure 8. Enlarged power consumption segment of MSQPS

In Figure 8 one interval can be seen in detail. The transmission peaks (9) are smaller compared to all other schemes due to the small number of necessary bytes per transmission. Therefore, the peak for the memory transfer is also a bit smaller. The key exchange needs 3.33 s and is slightly faster than IKDM.

3.5 Key Revocation and Renewal Protocol

ECC was used in KRRP for the asymmetric part of the key exchange. The elliptic curve secp256r1 has been used, because the proposed secp160r1 is not recommended anymore [1]. The key exchange is done in two steps. First, the public key is exchanged. Subsequently, a session key is generated via ECC. The total determined energy consumption is 211.11 mJ. The key exchange needs in this scheme 2.71 s.

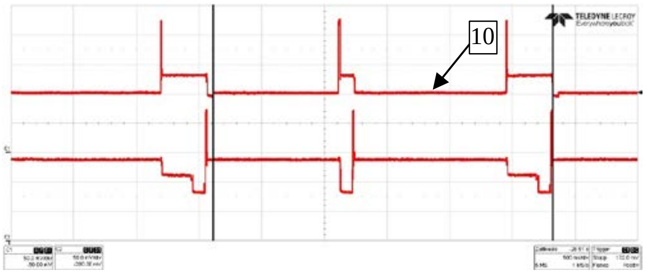

Figure 9. Complete power consumption of KRRP

The power consumption can be seen in Figure 9. The preparation and the first transmission is short and has nearly no additional computation effort compared to the Ping example because just the public key is transferred. The second preparation (10) takes much more time due to the session key and a nonce encryption through ECC. The second node has to decrypt the session key as well as the nonce and has to encrypt the nonce for the initiator node with the session key. This must happen before the node starts the preparation to transmit the encrypted nonce. Receiving the nonce, the initiator node has to decrypt it and to check if it is the correct one.

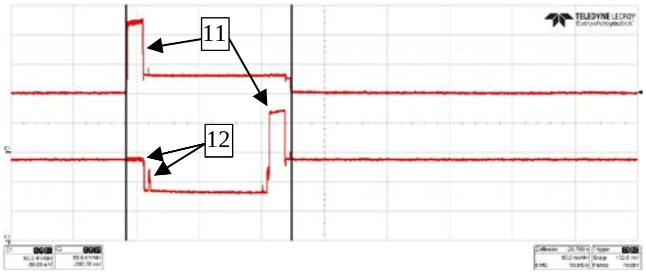

Figure 10. Enlarged power consumption segment 1 of KRRP

Figure 10 shows the current consumption of the public key transmission in detail. The transmission time is higher (11) than the transfer times of the other schemes due to the length of the public key. Therefore, the copy of the received data into the memory needs two time intervals for copy (12).

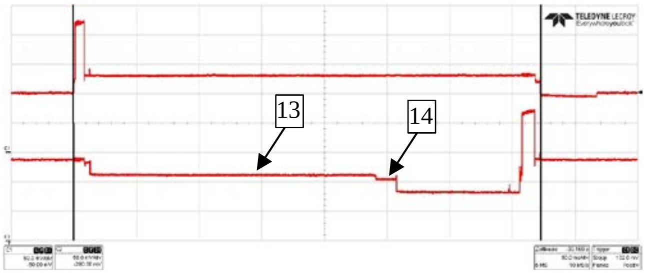

Figure 11. Enlarged power consumption segment 2 of KRRP

Figure 11 shows the asymmetric transmission of the session key and the increased computational effort of the asymmetric computation (12). The decryption takes much more time compared to the symmetric encryption (13). The encryption starts when the current curve slightly decreases.

3.6 Discussion of the Results

Table 1. Energy Consumption of all proposed schemes

Proposed scheme	Energy consumption	Time for key exchange
LAS	141.67 mJ	1.83 s
IKDM	256.88 mJ	3.35 s
MSQPS	261.42 mJ	3.33 s
KRRP	211.11 mJ	2.71 s

Table 1 shows the total energy consumption and the needed time of each key management protocol which was measured on the CC1350-LaunchPad boards. LAS, IKDM, MSQPS and KRRP are good schemes for energy efficient key management in WSNs. LAS is the best choice for WSNs where energy is the greatest issue. It is easy in implementation but it has the issue of symmetric key distribution. IKDM is a good choice for cluster based WSNs because it provides a good security level with a low energy consumption and low computation time. MSQPS is not recommended since it does not use standardized algorithms and confidentiality as well as integrity cannot fully be guaranteed. The total energy consumption of MSQPS is higher than IKDM on nearly the same computing time. The reason could be that for IKDM a crypto library developed for Cortex-M3 processors was used and the algorithms of MSQPS are not optimized for Cortex-M3. If security is the highest goal in a WSN, KRRP is the best choice. It is the only protocol which provides authenticity. Furthermore, the used algorithms and protocols of KRRP are very strong. KRRP has a higher energy consumption compared to LAS. IKDM and MSQPS need more energy than KRRP. This is due to the complexity of these schemes for handling cluster-based WSNs. Table 2 summarizes the comparison between the related schemes.

Table 2. Comparison of the selected schemes

Scheme	Confi-	Integ-	Avail-	Auth-	Fresh-
LAS	(x)	x	x	-	x
IKDM	(x)	x	x	-	x
MSQPS	(x)	(x)	x	-	x
KRRP	x	x	x	x	x

The comparison is based on the type of cryptographic algorithms used, the type of cryptographic technique and the security goals defined in chapter 2. The security goals marked with (x) are critical to be achieved by the schemes and needs additional effort to guarantee them. LAS and IKDM need a secure deployment of the initial master key to reach confidentiality. MSQPS needs a secure environment for achieving confidentiality and integrity during key exchange because it is not robust against sniffing attacks in this phase. In summary, it can be stated that there is no protocol suitable for all use cases of WSNs. Therefore, it makes sense to select protocols according to the application and to combine different protocols if necessary to get the needed security goals.

4. CONCLUSION AND FUTURE WORK

WSNs are an important technology for the industrial environment and IoT. WSN can be used to transfer data easily over long distances and connect devices which are hard to access via wire. Especially sensitive data needs a secure connection between all WSN devices. Therefore, key management for WSNs is one of the major critical issues that have been addressed through several papers. The low resources of the devices in a WSN allow no standardized techniques as used in IP-based communication. This paper provided a comparison of four different efficient key management techniques: LAS, IKDM, MSQPS and KRRP. The location and size of the network is important in determining which scheme should best be used. In addition, power-efficient software is supposed to be used on assembler-optimized software. This paper also demonstrates that the usage of asymmetric cryptography does not always result in a higher energy consumption. A combination of ECC for authentication and symmetric cryptography for standard data transfer is useful to combine the advantages of both.

For symmetric and asymmetric methods, especially when using elliptic curves, there may be further problems in the near future. Commonly used asymmetric methods today are vulnerable to attacks via quantum computers [8]. For this, longer keys or new algorithms must be used, which will lead to an increase in energy consumption. As quantum computer technology becomes more widely used, new algorithms for WSN must be deployed. This results in the need of more research in efficient methods of key management and key exchange.

5. ACKNOWLEDGMENTS

This research work has been supported by the research projects no. 03FH016IA5 and 13FH645IB6 of the German Federal Ministry of Education and Research and by the Ministry of Education, Youth and Sports of the Czech Republic under grant No. LO1506.

6. REFERENCES

[1] BSI. 2018. Cryptographic Mechanisms: Recommendations and Key Lengths. *BSI – Technical Guidline BSI TR-02102-1*

[2] Chattopadhyay, S., Turuk, A.K. 2012. A Scheme for Key Revocation in Wireless Sensor Networks. *IJACECT International Journal on Advanced Computer Engineering and Communication Technology, 1* (2), 16-20.

[3] Cheng, Y. and Agrawal, D.P. 2007. An Improved Key Distribution Mechanism for Large-Scale Hierarchical Wireless Networks Key Distribution. *Ad Hoc Networks, 5* (1), 35–48. DOI= https://doi.org/10.1016/j.adhoc.2006.05.011

[4] Delgado-Mohatar, O., Fúster-Sabater. A. and Sierra, J.M. 2011. A light-weight authentication scheme for wireless sensor networks, *Ad Hoc Networks*, *9* (5), 727-735. DOI= https://doi.org/10.1016/j.adhoc.2010.08.020

[5] García-Hernández, C.F.; Ibargüengoytia-González, P., García-Hernández, J., and Pérez-Díaz, J.A. 2007. Wireless Sensor Networks and Applications: a Survey. *IJCSNS International Journals of computer science and network security, 7* (3), 264-273.

[6] Ghosel, A. and DasBit, S. 2015. A lightweight security scheme for query processing in clustered wireless sensor networks. *Computers & Electrical Engineering, 41*, 240-255. DOI= https://doi.org/10.1016/j.compeleceng.2014.03.014

[7] Ghosel, A., Halder S., Sur S., Dan, A. and DasBit, S. 2010. Ensuring basic security and preventing replay attack in a query processing application domain in WSN. In *Computational Science and Its Applications - ICCSA*, (Fukuoka, Japan), Springer-Verlag, 321–335. DOI= https://doi.org/10.1007/978-3-642-12179-1_28

[8] Heigl, M., Schramm, M. and Dalibor, F. 2019. A Lightweight Quantum-Safe Security Concept for Wireless Sensor Network Communication. *PerCom Workshops-IEEE International Conference on Pervasive Computing and Communications Workshops*, (Kyoto, Japan), IEEE. ISBN 978-1-5386-9150-2

[9] Maletsky, K. 2015. RSA vs ECC comparison for embedded systems. Atmel. http://www.atmel.com/images/atmel-8951-cryptoauth-rsa-ecc-comparison-embedded-systems-whitepaper.pdf

[10] Mansour, I., Chalhoub, G. and Lafourcade, P. 2015. Key Management in Wireless Sensor Networks. *Journal of Sensor and Actuator Networks, 4* (3) 251-273. DOI= https://doi.org/10.3390/jsan4030251

[11] Suraj, S. and Jena, S. 2011. A survey on secure hierarchical routing protocols in wireless sensor networks. In *International Conference on Communication, Computing & Security-ICCCS*, (Odisha, India), ACM, 146-151. DOI= https://doi.org/10.1145/1947940.1947972

[12] Zhou, Y., Fang, Y. and Zhang, Y. 2008. Securing wireless sensor networks: a survey. *Communications Surveys & Tutorials IEEE, 10* (3), 6-28. DOI= https://doi.org/10.1109/COMST.2008.4625802