On the Energy Consumption of Quantum-Resistant Cryptographic Software Implementations Suitable for Wireless Sensor Networks

Michael Heigl[12], Laurin Doerr[12], Martin Schramm[2] and Dalibor Fiala[1]

[1]*Department of Computer Science and Engineering, University of West Bohemia, Pilsen 301 00, Czech Republic* [2]*Institute ProtectIT, Deggendorf Institute of Technology, 94469 Deggendorf, Germany*

Keywords: Wireless Sensor Networks, Post-Quantum Cryptography, Energy Consumption

Abstract: For an effective protection of the communication in Wireless Sensor Networks (WSN) facing e.g. threats by quantum computers in the near future, it is necessary to examine the applicability of quantum-resistant mechanisms in this field. It is the aim of this article to survey possible candidate schemes utilizable on sensor nodes and to compare the energy consumption of a selection of freely-available software implementations using a WSN-ready Texas Instruments CC1350 LaunchPad ARM® Cortex®-M3 microcontroller board.

1 INTRODUCTION

The set consisting of base stations and multiple tiny autonomous devices interconnected via an ad-hoc communication is called WSN. Nowadays, they have permeated many technological areas mainly driven by the Internet of Things (IoT). The sensor nodes are composed of a microcontroller, a radio transceiver, a power supply (typically battery-powered), memory and one or more sensors to collect data from the immediate environment and transmit it wirelessly. Application areas are boosted by the IoT ranging from simple measurements of environmental data such as temperature or smoke towards IoT-enabled Smart Cars (Bartolomeu et al., 2018).

The ever-increasing and more advanced attack capabilities and strategies pose an enormous challenge and demand IT-security for WSN in the near future. The sophistication of attacks utilizing distributed, cloud or even quantum computation (Sen, 2013) forces classical IT-security to change into sustainable cyber resilience. However, the application of viable cryptographic schemes vital for holistic security solutions are often seen as an undesirable overhead cost and therefore neglected in the design of WSN. Due to the inherent resource limitations in terms of memory size, processing speed as well as energy consumption, attacks towards WSN are manifold (Costa et al., 2017). Because of the open communication nature of the wireless radio channel, attackers can easily eavesdrop on, intercept, inject and forge the exchanged information. Especially when it comes to the transmission of confidential information, for instance measurements in the medical or military area, cryptographic schemes are inevitable to preserve the protection goals confidentiality, integrity and authenticity. The application of these schemes on the other side still imposes several limitations since the known resource issues and other constraints presented in (Sen, 2013) inhibit the deployment in WSN. Furthermore, with the advent of quantum computers, quantum computation will have a tremendous impact on a major field of cryptography not only erupting WSN-security. It has the potential to break or weaken asymmetric schemes including the EllipticCurve Diffie-Hellman (ECDH) key exchange often applied in the embedded domain.

Thus, in this work possible state-of-the-art cryptographic schemes are proposed which are resistant to the power of quantum computers. A selection of freely-available software implementation candidates are integrated on a WSN-ready microcontroller board and evaluated in terms of their applicability in WSN. The remainder of this paper is structured as follows: Section 2 discusses the threats from quantum computers towards cryptographic schemes. A state-of-the-art review on cryptographic mechanisms with regard to quantum-resistance for WSN is provided in Section 3. Details on the measurement setup and integration of a selection of freely-available software implementation candidates on the Texas Instruments CC1350 LaunchPad microcontroller board are presented in Section 4. The evaluation in Section 5 deals with energy consumption measurements with respect to fundamental

This is a preprint of “Heigl, M., Doerr, L., Schramm, M., & Fiala, D. (2019). On the Energy Consumption of Quantum-resistant Cryptographic Software Implementations Suitable for Wireless Sensor Networks. *Proc. 16th International Joint Conference on e-Business and Telecommunications (SECRYPT)*, Prague, Czech Republic, pp. 72-83.” https://doi.org/ 10.5220/0007835600720083

cryptographic components and discusses their computational and communication cost. A short conclusion and a glance at the future work of the ongoing research work finalizes the article in Section 6.

2 QUANTUMNERVOUSNESS

Quantum computing is founded on the microscopic physical phenomena of quantum mechanics. Instead of using bits with observed states 0 and 1 on contemporary computers, quantum computers are based on quantum bits (qubits). Qubits are particles that can exist in both states simultaneously, by the effect of superposition, and are able to exploit true parallelism due to quantum entanglement (Jozsa, 1997). In the era of having first viable quantum computers, the socalled *quantum nervousness* begins and represents a new type of threat for cryptography in general.

Shor's algorithm (Shor, 1997) makes use of the quantum Fourier transform to extract the required information from the superposition of quantum states which allows to break all contemporary asymmetric cryptography such as RSA, DSA or EllipticCurve Cryptography (ECC) based schemes. Especially ECC is widely deployed in IoT-environments due to the more efficient arithmetic, low memory usage, shorter key sizes and lower CPU consumption compared to RSA or DSA (Mani and Nishamol, 2013). Even if asymmetric cryptography with adequate key sizes (BSI, 2018) is considered very safe today, the underlying mathematical problems such as efficient integer factorization e.g. in RSA or the calculation of the discrete logarithm e.g. in ECC can be solved with Shor's algorithm in a reasonable amount of time if a powerful quantum computer can be utilized. Even if such efficient machines are currently far from being practically feasible, the NIST states that a quantum computer capable of breaking RSA2048 or ECC-256 in a matter of hours could be built by 2030 (NIST, 2016).

A breakthrough has recently been announced by Google regarding its 72-qubit capable Bristlecone paving the way for large scale quantum computers (Kelly, 2018). Those will also affect symmetric cryptography such as AES. Even if Grover's algorithm (Grover, 1996) has been initially intended to search unsorted databases, it can be applied to crack symmetric ciphers. However, the impact is far less than with Shor's algorithm since currently it is assumed that doubling the key length mitigates its harm. According to the authors of (Amy et al., 2016), cryptographic hash functions are not affected by quantum computers. They state that both SHA-256 and SHA3-256 need around 2166 logical qubit cycles to be cracked and even if the quantum correction is handled by hardware, running at a few million hashes per second, Grover's algorithm would still need about 1032 years.

3 CRYPTOGRAPHYFORWSN

The protection of communicated data in WSN to achieve the stated security goals is mainly founded on hash functions, symmetric-key and asymmetric-key algorithms composed of the following crucial components: *message authentication*, *message encryption / decryption*, *key exchange*, *digital signature*. Especially with the threat of quantum computers towards asymmetric cryptography, practical alternatives for contemporary schemes in WSN such as ECDH or DSA must be researched.

Post-quantum cryptography (PQC) refers to cryptographic schemes that can function on classical non-quantum machines but promise to withstand the impact of Shor's or Grover's algorithm running on quantum-accelerated computers. Several different families of PQC are presented in (Bernstein et al., 2009) including hash-based, codebased, lattice-based, multivariate-quadratic-equations and secret-key cryptography. Supersingular ellipticcurve isogeny cryptography is much closer to classical ECDH-schemes, but it is very young and not well researched (Bogomolec and Gerhard, 2018).

Recommendations for cryptographic schemes to be used in the quantum-era, not intended for embedded domains, are provided by (PQCRYPTO, 2015). Even if some PQC software implementations aimed for low-cost devices exist (Xu et al., 2018), a major demerit is the poor resource-efficiency especially in terms of communication overhead of most of the postquantum schemes compared to contemporary cryptography which makes them hardly applicable on resource-constrained devices. Another uncertainty with PQC is that there could be a currently unknown algorithm that breaks even such schemes in the near future. Thus, a combination with contemporarily used mechanisms could be a possible tradeoff. An example is Google's CECPQ1 which is a key exchange that concatenates the results of an X25519 and NewHope key exchange. X25519 is a

Diffie-Hellman-based key exchange applying Curve25519.

The following sections discuss the cryptographic key components with respect to the application in WSN by stating contemporary applied schemes and possible PQC-alternatives. With respect to the security level, schemes providing 128-bit will be considered, since the 112-bit security level should be phased out by 2031 (NIST, 2016). For each key component respective candidate software implementations applicable on resource-constrained devices are identified, later compared and evaluated.

3.1 Message Authentication

Message Authentication Codes (MAC) are used to achieve message integrity and authentication. MACs are used to identify unauthorized and corrupted messages and can be either defined over symmetric ciphers as modes of operation or, more commonly, on one-way hash functions such as HMACs. Regarding the threat of quantum computers, HMAC is demonstrably resistant (to 128-bit security level) even if an adversary can obtain the MAC of chosen messages, under weak hypothesis for SHA-256 (Bellare, 2015). Alternatives for assembly-optimized HMACs targeted for WSN environments are for instance the lightweight MAC Chaskey proposed in (Mouha et al., 2014), the LMAC of (Chowdhury and DasBit, 2015), the LightMAC (Luykx et al., 2016) or an assembly-optimized Poly1305. Recommended by the PQCRYPTO project are the GCM mode of AES using a 96-bit nonce and an 128-bit authenticator and Poly1305 (PQCRYPTO, 2015).

3.2 Message Encryption / Decryption

Symmetric cipher schemes are mainly used for message encryption / decryption and can be generally divided into block ciphers (e.g. AES, RC5, RC6, Skip Jack, HIGHT, BSPN) and stream ciphers (RC4, Sosemanuk, HC-128, Dragon, LEX or Salsa20/ChaCha20) (Karuppiah and Rajaram, 2012; Zhang et al., 2012; Meiser et al., 2008). The requirements of symmetric schemes with respect to WSN are energy consumption, program memory (storage), temporary memory (RAM) and execution time (Karuppiah and Rajaram, 2012). In (Meiser et al., 2008) various schemes have been implemented in assembler and ported to an 8-bit AVR microcontroller. Considering the encryption speed, all stream ciphers except Salsa20 outperformed AES and in terms of memory needs Salsa20 and LEX are almost as compact as AES. For high throughput requirements, Sosamuk turned out to be the most suitable cipher if its considerable higher memory needs can be tolerated. The authors of (Zhang et al., 2012) compared different block and stream ciphers in software and suggest that utilizing AES achieves better performance for a wide range of channel qualities and provides significant improvement in energy efficiency compared to other schemes. The PQCRYPTO project recommends the AES-256 and Salsa20 with a 256-bit key. Assembly-optimized Salsa20 or the assembly-supported AES as shown in (Schwabe and Ko, 2017) seem to be promising candidates for quantum-resistant WSN-devices. Both providing a 256-bit security level which guarantees sufficient security for encryption / decryption even in the presence of moderate quantum computers.

3.3 Key Exchange

Key management in WSN is a systematic process consisting of the key deployment/pre-distribution, the key agreement as an authenticated key exchange, the member/node addition, the member/node eviction and the key revocation (Kumaran et al., 2016). The obtained shared secret can then be used to derive a symmetric message encryption key. Protocols for key management in WSN are discussed in (Sen, 2013) and for key agreements in (Kumaran et al., 2016). Those comprise methods using symmetrickeys, asymmetric-key agreements as well as ones utilizing a trusted third party. Symmetric schemes require less computing power but suffer from the issue of an initial key exchange over an insecure channel. In addition to this, the problem occurs when the number of nodes in a WSN changes dynamically. Using a trusted third party is also unattractive since large sensor networks are characterized by multi-hop communication and demand more energy compared to asymmetric solutions. Most widely deployed in WSN are ECDH variants which allow two parties, each having an elliptic-curve key pair, to establish a shared secret over an insecure channel. However, many implementations, e.g. (Mani and Nishamol, 2013), are based on curves such as secp160r1. According to the SEC 2 (Certicom Research, 2000) standard they do not guarantee adequate security and referring to the German BSI (BSI, 2018), the use of mechanisms that achieve a security level of at least 120 bits is recommended for the prediction period beyond the end of 2022.

PQC key exchange mechanisms, in the following including key encapsulation mechanisms, based on lattices, e.g. Frodo, Kyber, Round5, some of them implemented in the Open Quantum Safe project (Open Quantum Safe Project, 2018), are not optimized for the usage in WSN (ETSI, 2017). For SIDH or SIKE currently only an implementation for the ARM Cortex-A15 and Cortex-A8 has been optimized utilizing the NEON instruction set (Jalali et al., 2018; Seo et al., 2018). Even if a prominent PQC-ready example, NTRU, could be used for a key exchange as proposed in (Lei and Liao, 2013), it is associated not only with a large computational cost but also with a high communication overhead due to large key sizes. Meanwhile an implementation of the NTRUEncrypt public-key cryptographic scheme exists, which can be used for a NTRU key exchange, optimized for a Cortex-M0 microcontroller, but no source code is publicly available (Guillen et al., 2017). It is not only suitable for use in battery-operated devices but also patents on the cryptosystem expired in 2017 or will expire in 2021 respectively. With various generic and platform-specific optimization, promising candidates for an IoT-enabled post-quantum key exchange guaranteeing 128-bits of post-quantum security are NewHope (Xu et al., 2018) or Saber which both have been ported to ARM Cortex-M architectures (Alkim et al., 2016; Karmakar et al., 2018). In addition, NewHope and Saber are among the 17 second-round candidates for key exchange algorithms of NIST's PQC standardization process (NIST, 2019). Even if according to (Alkim et al., 2016) NewHope outperforms ECDH implementations in terms of execution time, the large number of bytes to be exchanged in both algorithms must be considered in terms of timing and energy consumption.

3.4 Digital Signature

Digital signatures based on public key cryptography are necessary in order to authenticate for instance the key exchange process. In contrast to message authentication and encryption / decryption schemes, the requirements regarding memory or execution time are smaller since signing and verifying messages in WSN is often only used for partner identification during the initial key exchange or key renewal process. Mainly applied in WSN are ECDSA variants due to their efficiency (Ateniese et al., 2017). However, since also belonging to asymmetric schemes founded on the discrete logarithm problem, the mentioned approaches provide no protection in the era of quantum computers. Possible PQC schemes with regard to (NIST, 2019) based on lattices are Falcon, Dilithium and qTesla. Multivariate-based signature schemes are GeMSS, LUOV, MQDSS or Rainbow. However, recent work shows the vulnerability of lattice-based signature schemes to differential fault attacks (Bruinderink and Pessl, 2018). Targeted to WSN, the PQM4 library provides candidate implementations for the Cortex-M4 (PQM4, 2019) and (Guneysu et al., 2018) focuses on implementa-¨ tions of GLP, BLISS and Dilithium for the M4 which do not claim protection against side-channel attacks. The work in (Margi et al., 2017) compares different post-quantum signature schemes on an 128-bit security level. The most prominent ones are either based on lattices such as NTRUSign (Driessen, 2007), BLISS (Howe et al., 2015) or on hash functions such as extended Merkle's signature scheme (XMSS) with similar approaches for IoT-environments (Pereira et al., 2016) or ARMed SPHINCS (Hulsing et al.,¨ 2016). Compared for instance to lattice-based PQCschemes, hash-based signatures are well understood and the security relies on the underlying and thus very well studied security of hash functions. XMSS and SPHINCS are recommended by the PQCRYPTO project. Especially ARMed SPHINCS based on SPHINCS-256 is a promising implementation for the use on embedded systems. It is a high-security post-quantum stateless hash-based signature scheme and provides an effective replacement for signatures by combining XMSS and other techniques to overcome the drawback of statefulness of XMSS. SPHINCS+ is furthermore among the 9 secondround candidates for digital signatures of NIST's PQC standardization process (NIST, 2019). Even if the most advanced approaches for practical hash-based signatures SPHINCS-256 and the multi-tree variant XMSS^{MT} are well studied hash-based schemes, recent work shows different attacks on them (Kannwischer et al., 2018).

4 MEASUREMENTSETUP

The evaluation of various freely-available cryptographic software implementations is performed using LaunchPad boards equipped with a 32-bit ARM Cortex-M3 processor. The scope of the evaluation are measurements regarding the energy consumption such that a statement on the applicability of quantumresistant schemes for WSN can be made. Thus, in the case of asymmetric schemes, especially threatened by Shor's algorithm, a comparison with contemporary mechanisms in terms of computational and communication cost is conducted. For programming the boards via the onboard debugger, the Code Composer Studio in Version 8.0.0.00016 has been used. As basis projects, *rfEasyLinkEchoTx* and *rfEasyLinkEchoRx* are used from the Texas Instruments TI Resource Explorer which demonstrate the usage of the EasyLink API. A DC voltage source providing V_{cc} = 3.3 V is used to source one board with an interconnected lowside shunt resistor of R = 10 Ω. A 1 GHz oscilloscope with a maximum of 20 GS/s is used together with a passive probe to measure the voltage drop over the shunt in order to compute the energy consumption. The oscilloscope settings are DC coupling with 1 MΩ, an ERES noise filter +3bits, a sinx/x interpolation, a trigger slope with rising edge and time settings with maximum sample points of 500 kS/s with realtime sampling and a delay of 1.6 s. In order to only measure the energy consumption of the CC1350, the onboard debugger needs to be disconnected.

The reference project was extended with the relevant source files of each cryptographic software implementation. The task functionality for transmitting and receiving was disabled for the following measurements since these where easier without the power management of TI-RTOS. Therefore, for each library, a dedicated test function is executed after initialization performing the relevant cryptographic algorithm. Each oscilloscope diagram shows two curves. For signal C1, a voltage drop at the beginning based on the switching off of an onboard LED signals the start of the respective cryptographic functions. After the start and between each relevant functions some delays of simple *nop*-commands in assembler are introduced in order to measure each cryptographic component in the zoomed curve Z1 of signal C1.

During the idle phases, a voltage of approximately 76 mV is measured resulting in a current of 7.6 mA.

5 EVALUATION

The following candidates of freely-available software implementations have been selected for evaluation on the candidate WSN board with respect to the cryptographic key components of Section 3. Even if ARM Cortex-M0 and M4 microcontrollers are very popular for realizing IoT-applications (Karmakar et al., 2018), the evaluation using a CortexM3 whose instruction set is only downwards compatible excludes the utilization of M4-optimized implementations to be more restrictive in terms of resources for WSN. For the sake of comparison, even if not quantum-resistant, some ECC-based implementations for asymmetric schemes have been chosen. For evaluating different MAC candidates, the *LightMAC*[1]implementation based on the publication (Luykx et al., 2016), *Chaskey*[2] and HMAC as well as Poly1305 both taken from the embedded-optimized cifra [3] library donated as *HMAC SHA-256$_{cifra}$* and *Poly1305$_{cifra}$* have been utilized. Comparative energy consumption results on different processor architectures can be found in (Chowdhury and DasBit, 2015). For symmetric message encryption / decryption, the *tiny AES*[4], the assembly-supported AES on ARM Cortex-M3/M4 *aes-armcortexm*[5] library from the paper (Schwabe and Ko, 2017), the *Salsa20$_{cifra}$*[3] taken from the cifra library and the assembly-optimized *chacha20$_{ARM}$* [6] library of the cipher ChaCha20 have been tested. For comparing with implementations on a 133 MHz StrongARM processor refer to the results of (Großschadl et al., 2007).¨ For evaluating a software-based key exchange process between two boards, the SafeCurve (Bernstein and Lange, 2014) implementation of an X25519 *curve25519-donna*[7] in ANSI C, the assembly-supported *ECDH$_{micro-ecc}$*[8] library having an ECDH function and the post-quantum

[1] https://github.com/SebMertz/LightMAC

[2] https://mouha.be/chaskey/

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

[4] https://github.com/bricke/tiny-AES-C

[5] https://github.com/Ko-/aes-armcortexm

[6] https://gitlab.science.ru.nl/mneikes/armchacha20/tree/master

[7] https://github.com/agl/curve25519-donna

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

algorithms *Saber_{ARM}*[9] and *NewHope_{ARM}*[10] optimized for ARM Cortex-M0 have been selected. Comparative results on the energy consumption utilizing different processor architectures can be found in (Moon et al., 2016; Großschadl et al., 2007;¨ Bianchi et al., 2010). For the testing of signature generation and verification, the assembly-supported *ECDSA_{micro-ecc}*[8] library having an ECDSA function, the C implementation *lamport-OTS*[11] of the cryptographic Lamport-style One-Time-Signatures (OTS), the implementation *SPHINCS_{Port.}*[11] with the aim to port *ARMed SPHINCS* on a Cortex-M3 based on the source code of the paper (Hulsing et al., 2016) and¨ the *Shorter Merkle Signatures*[12] implementation of the paper (Pereira et al., 2016) have been chosen. For comparing with implementations on other processor architectures refer to the results of (Rehana et al., 2010; Rehana et al., 2012; Ateniese et al., 2017).

5.1 Message Authentication

For each message authentication evaluation 64 bytes of input data have been used. In Figure 1 an exemplary plot of the oscilloscope output for the HMAC SHA-256$_{cifra}$ implementation is shown. The upper curve (signal C1) of Figure 1 shows the measured voltage of one MAC computation on a single board and signal Z1 the zoomed square wave of the MAC signal in more detail. The average amplitude of the signal for only message authentication is approximately 528 µA and the duration 3.52 ms. Thus, the energy consumption only for message authentication is approximately 6.13 µJ. For the sake of comparison with the symmetric ciphers, the 64 byte results are converted to 16 bytes resulting in a message authentication taking approximately 880 µs and consuming about 1.53 µJ. Including the board energy consumption for the execution of the message authentication this results in approximately 23.6 µJ.

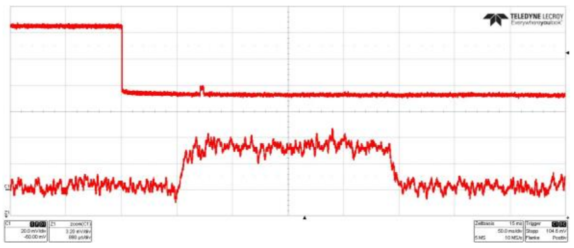

Figure 1: Measurement of HMAC SHA-256$_{cifra}$ message authentication

Table 1 summarizes the measurements for the MAC implementations. Information is represented in Table 1 as follows. The column labelled "Library" gives the name of the used message authentication library. The total time the message authentication takes for 64 bytes is provided with column "Time". The column "Energy" provides information about the energy consumption only of the message authentication without considering the total board energy consumption. Thus, column "Total" of Table 1 provides the total energy consumption of the tested schemes on the CC1350 board for an idle current of 7.6 mA computed for the execution time of each algorithm excluding delays. For the sake of comparison with the symmetric cipher schemes, the column "Total$_{16}$" lists the naively computed values for the total board energy consumption of generating message authentication codes from 16 bytes input data. The implementations of the widely accepted SHA-256 HMAC and Poly1305 of the cifra library perform quite similar in terms of energy consumption and execution time. The lightweight implementations Chaskey and LightMAC, however, outperform the others. Especially LightMAC is a good candidate to be applied for WSN in terms of energy consumption.

5.2 Message Encryption / Decryption

For AES-based libraries, AES-256 has been chosen since, if a moderate quantum computer utilizing Grover's algorithm is realized, it seems prudent to move away from 128-bit keys (Grassla et al., 2016). In Figure 2 an exemplary plot for the Salsa20$_{cifra}$ measurement is shown. The upper curve (signal C1) of Figure 2 shows the measured voltage of one

[9] https://github.com/KULeuven-COSIC/SABER/tree/master/SABER_ARM

[10] https://github.com/newhopearm/newhopearm

[11]https://github.com/m0jo/lamport-OTS

[11] https://github.com/AymericGenet/SPHINCSarduinodue

[12] https://github.com/puodzius/shorter-hashsignatures

encryption and decryption. The key setup and initialization vector setting function had no influence on the measured signal and thus has been neglected. Signal Z1 shows the zoomed square wave of the encryption / decryption signal in more detail. The average amplitude of the signal is approximately 460 µA and the duration 22.6 ms. Thus, the computed values for 16 bytes referring to one block in AES is approximately 88.28 µs. The energy consumption for the encryption of one block only is approximately 134 nJ.

Table 1: Summary of the measurements regarding the message authentication schemes

Library	Time (ms)	Energy (µJ)	Total (µJ)	Total$_{16}$ (µJ)
HMAC SHA-256$_{cifra}$	3.52	6.13	94.40	23.6
Poly1305$_{cifra}$	3.55	5.93	94.96	23.74
Chaskey	2.24	2.63	58.81	14.7
LightMAC	0.65	0.47	16.77	4.19

Table 2: Summary of the measurements regarding the symmetric cipher schemes

Library	Type	Time (µs) enc/dec	Energy (nJ)	Total (µJ)
tiny AES	AES-256 CBC	1290/8000	1610/3990	33.96/204.6
Salsa20$_{cifra}$	Salsa20	88.28/88.28	134/134	2.35/2.35
chacha20$_{ARM}$	ChaCha20	50.78/50.78	92.95/92.95	1.37/1.37
aes-armcortexm	AES-256 CTR	25.98/29.96	20.73/24.50	0.67/0.76

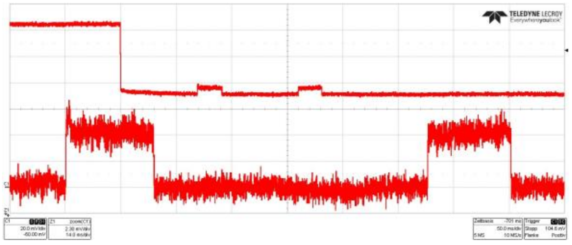
Figure 2: Measurement of Salsa20$_{cifra}$ encryption / decryption

Table 2 summarizes the measurements for the symmetric cipher schemes. For AES schemes, the column “Type” names the used AES bit sizes and the respective applied modes. In the column “Time” the total time for encryption (enc) and decryption (dec) is provided. It is clear that the aes-armcortexm library greatly outperforms tiny AES since it is an assemblysupported library presented in (Schwabe and Ko, 2017) for the ARM Cortex-M3 and M4. However, in terms of energy consumption tiny AES is competitive to the implementation presented in (Großschadl¨ et al., 2007). In summary, the encryption and decryption of this solution for one 128-bit block requires approximately 13.9 µJ. In contrast energy costs for both tiny AES modes are less than 10 µJ. However, in terms of execution time tiny AES is inferior. The ChaCha20 stream cipher achieves good performance which is almost competitive to the aes-armcortexm block cipher and slightly better than the Salsa20 implementation. In terms of energy cost both schemes outperform the implementation of (Großschadl et al.,¨ 2007) by a factor greater than 70 (for chacha20$_{ARM}$) and 150 (for aes-armcortexm). It must be noted that the chacha20$_{ARM}$ library is optimized for the CortexM0 whereas the aes-armcortexm library is aimed at the more powerful Cortex-M3 and M4, thus, a direct comparison is not possible with the used libraries.

5.3 Key Exchange

In Figure 3 an exemplary plot for the practical ARM Cortex-M0 ported NewHope implementation is shown. The upper curve (signal C1) of Figure 3 shows the measured voltage of the relevant key encapsulation computations on a single board. The random byte generation at the beginning of the algorithm has been neglected. Signal Z1 shows the zoomed square wave of the key exchange components in more detail. Those are not equal in terms of amplitude and duration as it is the case with other libraries. The average amplitude of the signal executing the key generation is approximately 465 µA and the duration 35.7 ms. The average amplitude of the signal executing one partners’ shared secret is approximately 515 µA and the length 54.8 ms. The average amplitude of the signal executing the other partners’ shared secret is approximately 294 µA and the duration 9.5 ms. The key exchange excluding the introduced delays and transmitting as well as receiving operations between two parties takes approximately 100 ms. The energy consumed by the key exchange yields approximately 0.15 mJ.

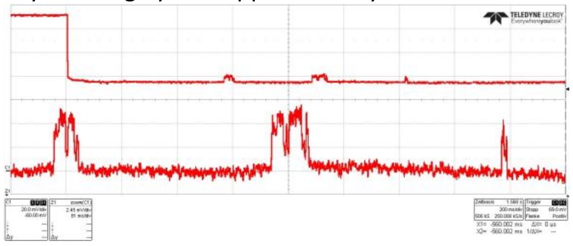

Figure 3: Measurement of NewHope$_{ARM}$ key exchange

It must be noted that in contrast to NewHope's three relevant key exchange computations necessary for the key encapsulation mechanism (key generation, encapsulation and decapsulation), ephemeral ECDH-based key exchange protocols are composed of four computationally expensive scalar multiplications. The security strength of the used key exchange schemes must be comparable. For this reason, to compare the elliptic curve based implementations with the PQC ones, elliptic curves offering 128 bit security were chosen. Thus, for the ECDH$_{micro-ecc}$ library the curve secp256r1 was utilized. Apart from the chosen elliptic curves from the SEC 2 standard, Curve25519 offers 128 bits of security as well and in contrast to the other libraries is a representative of a safe elliptic curve (Bernstein and Lange, 2014). Table 3 summarizes the measurements for the key exchange libraries each having a security strength of 128 bit. "Curve" names the used curves for the ECC-based libraries.

Curve25519 is not only considered a safe curve but also significantly faster for the use in ECDH operations. The plain execution time of a key exchange on a single board takes approximately 1.48 s. For this library the power consumption is the highest of all evaluated ones, but the energy and total consumption is moderate. It must be noted that the curve25519-donna library is not using any assembly and is comparable to the assembly-optimized ones. The key exchange with the secp256r1 curve of ECDH$_{micro-ecc}$ for instance takes approximately 1.2 s and is slightly more efficient in terms of power and energy consumption than Curve25519. NewHope$_{ARM}$ outperforms the successive libraries in all metrics. It is optimized in all performance-critical routines for constrained devices in ARM assembly. Because of this optimization, the key exchange lasts only approximately 100 ms and is more efficient than ECDH$_{micro-ecc}$ by a factor of 10. Saber is comparable to ECDH$_{micro-ecc}$ but inferior to NewHope's results. However, Saber$_{ARM}$ is CCA-secure in contrast to NewHope$_{ARM}$. A good and secure competitor to NewHope would be an assembly-supported curve25519-donna library. In (Dull et al., 2015) curve operations in assembly are¨ presented for different microcontroller architectures, which would significantly speed up the curve25519donna library. However, according to a comparison mentioned in (Alkim et al., 2016), the NewHope implementation still outperforms Curve25519 of (Dull¨ et al., 2015) by more than a factor of two. Following the example of CECPQ1, a combination of an assembly CCA-secure NewHope with an optimized X25519 would be a reliable solution for WSN.

Table 3: Summary of the measurements regarding the key exchange schemes

Library	Curve	Time (s)	Energy (mJ)	Total (mJ)
curve25519-donna	Curve25519	1.48 (4 x 370 ms)	4.05	41.17
ECDH$_{micro-ecc}$	secp256r1	1.20 (4 x 300 ms)	1.78	31.88
Saber$_{ARM}$	-	1.15 (292 ms / 390 ms / 472 ms)	1.51	30.44
NewHope$_{ARM}$	-	0.10 (35.7 ms / 54.8 ms / 9.5 ms)	0.15	2.66

5.4 Digital Signature

In this evaluation some of the less publicly available implementations for digitally signing and verifying on embedded devices are tested. In Figure 4 an exemplary plot for the ECDSA implementation of the micro-ecc library is shown. ECDSA$_{micro-ecc}$ was tested for signature generation and verification of 32 bytes based on ECC curve secp256r1 on the CC1350 board. The upper curve (signal C1) of Figure 4 shows the measured voltage of the key generation, signature generation and signature verification. Signal Z1 shows the zoomed square waves of the components in more detail. The average amplitude of the signals is approximately 415 µA. The execution time for key generation is about 313 ms, for signature generation about 364 ms and for signature verification about

340 ms. Thus, the energy consumption is approximately 429 µJ, 498 µJ and 466 µJ. Including the board energy consumption the cost is approximately 8.3 mJ, 9.6 mJ and 9.0 mJ.

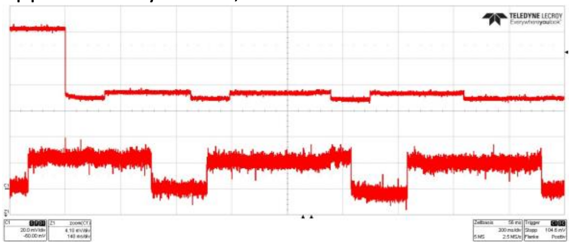

Figure 4: Measurement of ECDSA$_{\text{micro-ecc}}$ digital signature

The most promising implementation targeted for an ARM Cortex-M3 is ARMed SPHINCS presented in (Hulsing et al., 2016).¨ However, the provided source code is not directly applicable for the CC1350 since it is optimized for the STM32L1 sensor node. The portation SPHINCS$_{\text{Port.}}$ for the Arduino Duo board was used for the evaluation with the provided parameter set. Unfortunately, the verification function has not been ported so far and the library currently does only implement the plain SPHINCS-256 without any optimization. It must further be noted that SPHINCS has many parameters to adjust because of the composition of other schemes which makes it quite complex for a practical usage. The lamport-OTS implementation is for educational use and was tested for signature generation and verification of 32 bit. The weak hash function in the reference code has been replaced by the cryptographically secure SHA256 implementation [13]. The Shorter Merkle Signature implementation of the scheme presented in (Pereira et al., 2016) was tested for signature generation and verification of 32 bytes on the CC1350 board. Parameters have been adjusted because otherwise the execution would take too long. Those include the tree height with value 5, the *MSS K* with value 3 of the treehash algorithm BDS being used and the *WINTERNITZ W* parameter with value 2 for a signature size speed tradeoff. The parameter set allows the generation of 32 signatures and the first signature was used for signing and verifying. If the *n*-th signature is used, the process will take longer for larger *n*. The key generation process with larger parameter settings to create more signatures takes quite long but the signature generation and verification is significantly faster in comparison (minutes vs. seconds). Thus, it might be possible to perform the key generation and distribution to sensor nodes a priori.

Table 4 summarizes the measurements for signature generation and verification schemes. The total time the key generation, signature generation and verification takes is provided with the columns "Time" (key/sig/ver). The easiest form on which hash-based signatures are founded on are OTS whose key pair can only be used once. To sign multiple messages, Merkle trees composed of many OTS are proposed. The implementation lamport-OTS outperforms the other implementations but has severe drawbacks which makes it unpractical. Not only the one-time usage of a key pair is a problem but also if multiple keys are generated in advance, one must keep track of a large number of signatures. Therefore, the Shorter Merkle Signatures and the ARMed SPHINCS portation have been proposed. Due to the high timing cost for signing of about 200 s, SPHINCS$_{\text{Port.}}$ is not practically applicable. Even if the timing aspect for the public key authentication process is not critical, since it does not happen often, the signing time in the original paper of approximately 18.41 s is still too much. The reason behind the significant lack in performance lies in the cost for eliminating the state which makes SPHINCS to a stateless scheme. In contrast the stateful XMSSbased implementation Shorter Merkle Signatures of the paper (Pereira et al., 2016) is practically feasible. However, the key generation process for a larger number of signatures would take

Table 4: Summary of the measurements regarding the digital signature schemes

Library	Time (s) key/sig/ver	Energy (mJ) key/sig/ver	Total (mJ) key/sig/ver
ECDSA$_{\text{micro-ecc}}$	0.31/0.36/0.34	0.43/0.5/0.47	8.3/9.6/9
lamport-OTS	0.021/0.011/0.00032	0.043/0.0221/0.00065	0.57/0.29/0.0087
SPHINCS$_{\text{Port.}}$	7.5/200/-	11/295/-	199/5311/-
Shorter Merkle Sig.	4.1/2.9/0.098	4.1/0.2/0.06	107/5.14/1.57

[13] https://github.com/B-Con/crypto-algorithms

significantly longer and should be performed in advance.

5.5 Transceiving Messages

The energy efficiency for secure data transmission does not only rely on the computational cost of the used cryptographic algorithms but also on the communication cost. The latter can be split into the energy necessary not only for transmitting but also for receiving. This energy depends on the radio transmit power level, the distance between sending and receiving node and the time required for sending the message which is proportional to the message length. Even if the energy decomposition presented in (Bianchi et al., 2010) is mainly dominated by asymmetric cryptography, the main influence on the energy consumption over the battery lifetime is for transmitting and receiving in combination with the decryption of messages and/or the computation of MACs.

The rfEasyLinkEchoTx and rfEasyLinkEchoRx reference projects without the TI-RTOS have been modified in order to evaluate the communication cost on the CC1350 board. The curve of Figure 5 (signal Z1) shows the zoomed square wave of the transmitting (peak) and receiving (square wave) signal for 10 bytes transferred in detail. The used Phy setting is EasyLink Phy Custom. However, other Phy settings for instance *EasyLink Phy 50 kbps2gfsk* (Sub1G 50kbps data rate, IEEE 802.15.4g GFSK) yielded quite similar results in terms of energy consumption and transmitting as well as receiving time.

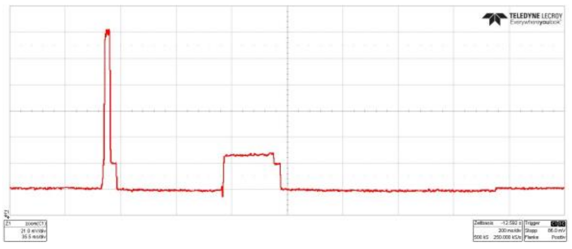

Figure 5: Current measurement of transmitting and receiving 10 bytes

In various measurements the transmit and receive time duration as well as the respective energy cost for different payload sizes has been determined. It must be noted that the payload size includes 1 byte for the length field and 1 byte for the destination address. Even if the maximum packet length supported by the TI 15.4 stack is 2047 bytes, the maximum payload size for transmitting and receiving on the CC1350 board using the reference projects is limited to only 125 bytes.

5.6 Key Exchange Overhead Costs

For the examined curve25519-donna based key exchange, with a public key size of 32 bytes, the resulting communication overhead is approximately 2·320 µJ for the transmission and 2·380 µJ for the reception. This results in a total of approximately 58 mJ for the key exchange regarding the computation and communication overhead including the board energy consumption. With the ECDH$_{\text{micro-ecc}}$ having a public key size of 64 bytes, the resulting overhead yields approximately 34 mJ. Some of the PQCschemes are according to (Margi et al., 2017) unsuitable for environments such as WSN since for instance transmitting a code-based public key would require about 1900 messages when applying IEEE 802.15.4 standard with a frame size of 127-byte. Something similar applies to the Saber/NewHope key exchange since one partner needs to transfer 992/1824 bytes and 1088/2048 bytes must be sent back. However, since this only applies to ideal conditions, it is suggested to not transmit packets of this size. Otherwise, it can lead to an increased power consumption in the case of retries when consuming a large air time slot. Thus, for an easy comparison based on the data from the message transmission and reception measurements, the packets for Saber and NewHope are fragmented using messages of up to 100 bytes. This yields a total board energy consumption for Saber of approximately 59 mJ and for NewHope of 56 mJ. Table 5 summarizes the key exchange measurements considering both the computational and the communication energy cost including the board consumption. “Comp. Time” names the duration the computation of the key exchange takes. The total time the communication of the key exchange takes is provided with column “Comm. Time”. The columns “Total$_{\text{Comp.}}$” and “Total$_{\text{Comm.}}$” provide information about the total energy consumption of the computation and communication of each key exchange algorithm including the total energy consumption of the CC1350 board.

Even if the NewHope implementation is outperforming the other libraries in terms of computational cost, the communicative overhead with the large amount of bytes to be transferred for a single key exchange is dominating the energy consumption and the execution time. It must be noted that the comparison is based on the calculation of fragmented messages. Increasing the message size could significantly reduce the communication overhead but, in contrast, makes the transmission more error prone. Considering both, the computational and communication energy consumption, the quantum-resistant implementations for Saber and NewHope show reasonable results compared to the ECC-based ones. Comparative work investigating energy cost and communication overhead of cryptographic protocols on different processor architectures is given in (Großschadl et al.,¨ 2007; Meulenaer et al., 2008).

6 CONCLUSIONANDFUTURE WORK

This work compares various freely-available software implementations of cryptographic schemes including post-quantum applicability targeted to the usage in WSN. Security goals are stated and the emerging attack vector by quantum computers are discussed. Cryptographic mechanisms including message authentication, symmetric encryption / decryption, key exchange and digital signature are reviewed and recommendations of candidates for WSN are provided. The evaluation mainly deals with measurements of the energy consumption including the discussion of computational and communication cost for candidate schemes on a dedicated WSN-ready reference platform. Chaskey and LightMAC seem to be promising candidates for message authentication in WSN. However, message authentication is more computationally complex than symmetric ciphers such that in a combined authenticated encryption an ARM assemblyoptimized AES-256-GCM or ChaCha-Poly1305 implementation is desirable. For symmetric encryption / decryption the aes-armcortexm implementation with highly optimized ARM Cortex-M3 assembly utilizing AES-256-CTR provides the best results followed by the Cortex-M0 optimized ChaCha20 implementation. The greatest threat of quantum computers is targeted to asymmetric cryptography. Thus, from the key exchange perspective, the NewHope for ARM seems a promising candidate which has been efficiently implemented on ARM architectures. Drawbacks on the other side are the missing protection against CCA which Saber's ARM implementation provides and the large amount of bytes to be transferred for NewHope which constitutes the main part of the energy consumption. Including the total computation and communication overhead NewHope and Saber seem applicable candidates for WSN compared to $ECDH_{micro-ecc}$ or curve25519-donna implementations providing the same pre-quantum security strength. The impact of the generation of random numbers on the energy consumption has been neglected in this article, but with respect to a real-world application, this must be considered in a future work.

The evaluated PQC signature implementations based on the research of (Pereira et al., 2016) and SPHINCS-256 are hash-based and neither seem applicable in real environments since XMSS schemes are stateful and SPHINCS-256 has a too high timing cost. Other schemes are currently either not feasible for the embedded use or still face threats by various attacks. Thus, more research in PQC-schemes is necessary in the future. With NIST's announcement on round 2 PQC candidates very recently only a limited number of optimized software implementations are available which still are mainly targeted for the Cortex-M4 architecture. Also shown with NewHope, dedicated assembly implementations for the CortexM families are necessary for the embedded domain since often additional hardware acceleration is not feasible due to financial reasons. Even if the CortexM family is backward compatible meaning that a M0 optimized assembly code is able to run on a M3 or M4, the instruction set capability is larger the more performant the processor resulting in a lower energy consumption. Various versions of implementations are desired aimed for the manifold existing embedded architectures. Porting or evaluating M4-optimized implementations, e.g. SIKE, Round5, Frodo, qTesla or Dilithium (PQM4, 2019) for less-powerful microcontrollers is part of future work. Similar to classical ECC, supersingular isogeny-based ECC seems a promising approach in terms of smaller key sizes and moderate signature sizes compared to other PQCschemes for the application in WSN. More research must be done regarding this field since e.g. SIDH is according to the authors of (Bogomolec and Gerhard, 2018) very young and not well trusted. However, the large number of recent publications and Microsoft Research projects (referring to (Jalali et al., 2018)) actively working on these algorithms underpin their importance in the future.

Table 5: Summary of the computational and communication energy cost of the tested key exchange schemes

Library	Comp. Time (s)	Comm. Time (s)	Total$_{\text{Comp.}}$ (mJ)	Total$_{\text{Comm.}}$ (mJ)
curve25519-donna	1.48	0.11	41.17	1.40
ECDH$_{\text{micro-ecc}}$	1.20	0.14	31.88	2.16
Saber$_{\text{ARM}}$	1.15	1.68	30.44	28.45
NewHope$_{\text{ARM}}$	0.10	3.15	2.66	53.59

ACKNOWLEDGEMENTS

This research work has been supported by the projects no. 16KIS0539 and 13FH645IB6 of the German Federal Ministry of Education and Research and the Ministry of Education, Youth and Sports of the Czech Republic under grant no. LO1506.

REFERENCES

Alkim, E., Jakubeit, P., and Schwabe, P. (2016). Newhope on arm cortex-m. *In International Conference on Security, Privacy, and Applied Cryptography Engineering*, pages 332–349.

Amy, M., Matteo, O. D., Gheorghiu, V., Mosca, M., Parent, A., and Schanck, J. (2016). Estimating the cost of generic quantum pre-image attacks on sha-2 and sha3. *Cryptology ePrint Archive, Report 2016/992*.

Ateniese, G., Bianchi, G., Capossele, A. T., Petrioli, C., and Spenza, D. (2017). Low-cost standard signatures for energy-harvesting wireless sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 16(3), 64.

Bartolomeu, P. C., Vieira, E., and Ferreira, J. (2018). IOTA feasibility and perspectives for enabling vehicular applications. *CoRR*, abs/1808.10069.

Bellare, M. (2015). New proofs for NMAC and HMAC: Security without collision resistance. *Journal of Cryptology*, Volume 28, Issue 4:844878.

Bernstein, D. and Lange, T. (2014). Safecurves: choosing safe curves for elliptic-curve cryptography. *https://safecurves.cr.yp.to*, Online; accessed 26Feb-2019.

Bernstein, D. J., Buchmann, J., and Dahmen, E. (2009). Post-quantum cryptography. *Springer, Berlin, ISBN 978-3-540-88701-0*.

Bianchi, G., Capossele, A., Mei, A., and Petrioli, C. (2010). Flexible key exchange negotiation for wireless sensor networks. *In Proceedings of the fifth ACM international workshop on Wireless network testbeds, experimental evaluation and characterization*, pages 55–62.

Bogomolec, X. and Gerhard, J. (2018). Postquantum secure cryptographic algorithms. *CoRR*, abs/1809.00371.

Bruinderink, L. G. and Pessl, P. (2018). Differential fault attacks on deterministic lattice signatures. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 21–43.

BSI (2018). Cryptographic mechanisms: Recommendations and key lengths. *BSI - Technical Guideline BSI TR-02102-1*.

Certicom Research (2000). Sec 2: Recommended elliptic curve domain parameters. *Standards for Efficient Cryptography*, http://www.secg.org/SEC2Ver-1.0.pdf.

Chowdhury, A. R. and DasBit, S. (2015). LMAC: A lightweight message authentication code for wireless sensor network. *2015 IEEE Global Communications Conference (GLOBECOM)*.

Costa, D. G., Figueredo, S., and Oliveira, G. (2017). Cryp-ˆ tography in wireless multimedia sensor networks: A survey and research directions. *Cryptography*, 1(1).

Driessen, B. (2007). Efficient embedded implementations of security solutions for ad-hoc networks. *Diploma Thesis, Ruhr-University Bochum*.

Dull, M., Haase, B., Hinterw¨ alder, G., Hutter, M., Paar,¨ C., Snchez, A. H., and Schwabe, P. (2015). Highspeed curve25519 on 8-bit, 16-bit and 32-bit microcontrollers. *Design, Codes and Cryptography*, https://cryptojedi.org/papers/#mu25519.

ETSI (2017). Cyber; quantum-safe key exchanges. *ETSI TR 103 570 - Technical Report*, https://www.etsi. org/deliver/etsi_tr/103500_103599/103570/ 01.01.01_60/tr_103570v010101p.pdf.

Grassla, M., Langenberg, B., Rotteler, M., and Steinwandt,¨ R. (2016). Applying grovers algorithm to aes: quantum resource estimates. *In International Workshop on Post-Quantum Cryptography, Springer*, pages 29–43.

Großschadl, J., Szekely, A., and Tillich, S. (2007). The en-¨ ergy cost of cryptographic key establishment in wireless sensor networks. *Proceedings of the 2Nd ACM Symposium on Information, Computer and Communications Security*.

Grover, L. (1996). A fast quantum mechanical algorithm for database search. *28th Annual ACM Symposium on the Theory of Computing*, page 212.

Guillen, O. M., Poppelmann, T., Mera, J. M. B., Bonge-¨ naar, E. F., Sigl, G., and Sepulveda, J. (2017). Towards post-quantum security for iot endpoints with ntru. *DATE*.

Guneysu, T., Krausz, M., Oder, T., and Speith, J. (2018).¨ Evaluation of lattice-based signature schemes in embedded systems. *In 2018 25th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, pages 385–388.

Howe, J., Poppelmann, T., O'neill, T., O'sullivan, M., and¨ Gueneysu, T. (2015). Practical lattice-based digital signature schemes. *ACM Transactions on Embedded Computing Systems (TECS)*, 4(3), 41.

Hulsing,¨ A., Rijneveld, J., and Schwabe, P. (2016). Armed sphincs. *Public-Key CryptographyPKC 2016, Springer, Berlin, Heidelberg*, pages 446–470.

Jalali, A., Azarderakhsh, R., and Mozaffari-Kermani, M. (2018). NEON SIKE: Supersingular isogeny key encapsulation on armv7. *In International Conference on Security, Privacy, and Applied Cryptography Engineering*, Springer, Cham:37–51.

Jozsa, R. (1997). Entanglement and quantum computation. *Geometric Issues in the Foundations of Science, Oxford University Press, ed. S. Huggett et. al.*

Kannwischer, M. J., Genet, A., Butin, D., Kramer, J., and¨ Buchmann, J. (2018). Differential power analysis of xmss and sphincs. *Cryptology ePrint Archive, Report 2018/673*.

Karmakar, A., Mera, J. M. B., Roy, S., and Verbauwhede, I. (2018). Saber on arm. *IACR Transactions on Cryptographic Hardware and Embedded Systems*, pages 243–266.

Karuppiah, A. B. and Rajaram, S. (2012). Energy efficient encryption algorithm for wireless sensor network. *International Journal of Engineering & Technology (IJERT)*, Volume 1, Issue 3.

Kelly, J. (2018). A preview of bristlecone, googles new quantum processor. *https://ai. googleblog.com/2018/03/a-previewof-bristlecone-googles-new.html*,
Online; accessed 26-Feb-2019.

Kumaran, U. S., Nallakaruppan, M. K., and Kumar, M. S. (2016). Review of asymmetric key cryptography in wireless sensor networks. *International Journal of Engineering and Technology (IJET)*, 8:859–862.

Lei, X. and Liao, X. (2013). Ntru-ke: A lattice-based public key exchange protocol. *IACR Cryptology ePrint Archive*, 718.

Luykx, A., Preneel, B., Tischhauser, E., and Yasuda, K. (2016). A mac mode for lightweight block ciphers. *In International Conference on Fast Software Encryption*, pages 43–59.

Mani, D. M. and Nishamol, P. H. (2013). A comparison between rsa and ecc in wireless sensor networks. *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2 Issue 3.

Margi, C. B., Alves, R. C. A., and Sepulveda, J. (2017). Sensing as a service: Secure wireless sensor network infrastructure sharing for the internet of things. *Open Journal of Internet of Things (OJIOT)*, Volume 3, Issue 1.

Meiser, G., Eisenbarth, T., Lemke-Rust, K., and Paar, C. (2008). Efficient implementation of estream ciphers on 8-bit avr microcontrollers. *In 2008 International Symposium on Industrial Embedded Systems*.

Meulenaer, G. D., Gosset, F., Standaert, F. X., and Pereira, O. (2008). On the energy cost of communication and cryptography in wireless sensor networks. *IEEE International Conference on Wireless and Mobile Computing WIMOB'08*, In Networking and Communications:580–585.

Moon, A. H., Iqbal, U., and Bhat, G. M. (2016). Authenticated key exchange protocol for wireless sensor networks. *International Journal of Applied Engineering Research*, Volume 11.

Mouha, N., Mennink, B., Herrewege, A. V., Watanabe, D., Preneel, B., and Verbauwhede, I. (2014). Chaskey: An efficient mac algorithm for 32-bit microcontrollers. *In International Workshop on Selected Areas in Cryptography*, pages 306–323.

NIST (2016). Report on post-quantum cryptography. *NIST Report NISTIR 8105*, https://nvlpubs.nist.gov/ nistpubs/ir/2016/NIST.IR.8105.pdf.

NIST (2019). PQC standardization process: Second round candidate announcement. *https://www.nist. gov/news-events/news/2019/01/pqcstandardization-process-secondround-candidate-announcement*, Online; accessed 26-Feb-2019.

Open Quantum Safe Project (2018). Software for prototyping quantum-resistant cryptography. *https:// openquantumsafe.org/*, Online; accessed 26Feb-2019.

Pereira, G., Puodzius, C., and Barreto, P. (2016). Shorter hash-based signatures. *Journal of Systems and Software*, Volume 116:95–100.

PQCRYPTO (2015). Initial recommendations of longterm secure post-quantum systems. *Horizon 2020 ICT-645622*, http://pqcrypto.eu.org/docs/ initial-recommendations.pdf.

PQM4 (2019). Post-quantum crypto library for the arm cortex-m4. *https://github.com/mupq/ pqm4*, Online; accessed 26-Feb-2019.

Rehana, Y., Ritter, E., and Wang, G. (2010). An authentication framework for wireless sensor networks using identity-based signatures. *10th International Conference on Computer and Information Technology*.

Rehana, Y., Ritter, E., and Wang, G. (2012). An authentication framework for wireless sensor networks using identity-based signatures: implementation and evaluation. *IEICE TRANSACTIONS on Information and Systems*, 95.1:126–133.

Schwabe, P. and Ko, S. (2017). All the aes you need on cortex-m3 and m4. *In International Conference on Selected Areas in Cryptography*, pages 180–194.

Sen, J. (2013). Security in wireless sensor networks. *CoRR*, abs/1301.5065.

Seo, H., Liu, Z., Longa, P., and Hu, Z. (2018). SIDH on arm: Faster modular multiplications for faster postquantum supersingular isogeny key exchange. *Cryptology ePrint Archive, Report 2018/700*.

Shor, P. (1997). Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM J. Comput.*, 26(5):1484–1509.

Xu, R., Cheng, C., Qin, Y., and Jiang, T. (2018). Lighting the way to a smart world: Lattice-based cryptography for internet of things. *CoRR*, abs/1805.04880.

Zhang, X., Heys, H. M., and Li, C. (2012). Energy efficiency of encryption schemes applied to wireless sensor networks. *Sec. and Commun. Netw.*, 5(7):789– 808.