In-depth energy analysis of security algorithms and protocols for the Internet of Things

Winderickx, J.; Braeken, A.; Singelee, D.; Mentens, N.

Journal of Cryptographic Engineering
https://doi.org/10.1007/s13389-021-00274-7

REGULAR PAPER

In-depth energy analysis of security algorithms and protocols for the Internet of Things

Jori Winderickx[1] · An Braeken[2] · Dave Singelée[3] · Nele Mentens[1,4]

Received: 3 October 2020 / Accepted: 25 September 2021

Abstract
Devices that populate the Internet of Things (IoT) are typically constrained with respect to energy consumption. When the data that are processed, stored and/or communicated by these devices need to be secured, low-energy security mechanisms have to be designed and implemented. Related work mainly concentrates either on low-energy security algorithms and protocols, or on low-energy wireless communication. However, it is important for system developers to take into account the overall energy consumption of the IoT system when making design choices. Therefore, this work presents an in-depth analysis of the energy consumption of IoT devices that provide end-to-end secure communication and digital signatures. The paper follows a granular approach, profiling and measuring each individual contribution to the overall energy consumption, including the computation of cryptographic operations as well as the wireless transmission of messages in cryptographic protocols. The paper also calculates the minimal time period of a secure communication session in order to minimize the energy impact of the session's setup phase and thus minimize the overall average power consumption. The goal of this work is to provide assistance in the selection of a suitable wireless communication standard and cryptographic cipher suite for building end-to-end secure IoT applications.

Keywords Data security · Embedded software · Energy consumption · Internet of Things (IoT) · Wireless communication · Wireless networks

1 Introduction

1.1 Motivation

Each year, low-power microcontrollers (MCUs) become more powerful with respect to performance, storage and memory capacity. Additionally, the energy consumption of these devices is reduced yearly thanks to technological and architectural enhancements. MCUs are mainly used in energy-constrained environments like Internet-of-Things (IoT) applications. They are typically combined with a range of sensors on a printed circuit board and a battery in a small package such that they can be placed in remote locations unsupervised. This means that MCUs should be able to operate independently for periods of time ranging from a couple of days to several years. For this reason, lightweight communication techniques are used, like LoRaWan [1] in low-power wide-area networks (LPWAN), or bluetooth low energy (BLE) in low-power wireless personal area networks (WPAN). However, when personal or company-critical data are being processed and transmitted,

This work was funded by the WearIT4Health project which is carried out under Interreg V-A Euregio Meuse-Rhine and is supported by the European Union and the European Regional Development Fund and with financial support of the province of Limburg—Belgium. This work was also supported by CyberSecurity Research Flanders with reference number VR20192203.

✉ Jori Winderickx
jori.winderickx@kuleuven.be

An Braeken
abraeken@vub.be

Dave Singelée
dave.singelee@kuleuven.be

Nele Mentens
nele.mentens@kuleuven.be ; n.mentens@liacs.leidenuniv.nl

1 ES&S and imec-COSIC, KU Leuven, Leuven, Belgium

2 Department of Industrial Engineering (INDI), Vrije Universiteit Brussel, Brussels, Belgium

3 imec-COSIC, KU Leuven, Leuven, Belgium

4 LIACS, Leiden University, Leiden, The Netherlands

Published online: 24 November 2021

security algorithms and protocols need to be deployed, consuming a significant amount of energy and thus reducing the battery lifetime.

1.2 Communication versus computation

In order to quantify the impact of security provisions in IoT devices, this paper concentrates on the energy consumption of security algorithms and protocols for end-to-end security and digital signatures. We divide the energy cost of a device into three parts: communication, application, and security. The communication cost involves the transport of packets from the IoT device to the remote end user and vice versa. The application cost involves the operation of sensors and related data processing algorithms. Finally, the security cost consists of the execution of security algorithms to provide confidentiality and authentication. While the application cost is specific to the use case, the communication and security cost can be generalized. A lot of research has, therefore, gone into the evaluation of one of these two costs. However, security and communication costs are tightly intertwined. Security is required to protect wireless communication, but communication is required to enable security. In this paper, we explore both the communication and computation cost of providing and using an end-to-end secured communication channel in an IoT setting using a granular approach. It is not the intention of this paper to provide accurate energy measurement results. Rather, the focus is on estimating the relative impact of end-to-end security on the total energy consumption of the application.

1.3 Minimal session duration

A security session typically contains a setup and a runtime phase. During the setup, entities can be authenticated, and, shared cryptographic material is established. This is used during the runtime phase to encrypt and/or authenticate the data. However, the runtime phase has a finite lifetime. After this period, a new session must be established. The longer a session is maintained, the higher the risk of the security session getting compromised. The most basic method to determine the maximum duration of a session is to calculate the maximum amount of data that can be encrypted by an encryption key. Another method is to estimate the risk of a successful side-channel attack in order to define the maximum lifetime of a session based on the number of traces an attacker can collect. Side-channel attacks exploit the information contained in, e.g., the power consumption [2], the electromagnetic emanation [3] or the timing behavior [4] of a device. The number of side-channel traces measured with the same key is typically proportional to the chance of a successful attack. Hence, in order prevent a successful attack, the lifetime of a session should be limited. While these two methods put an upper bound on the session duration, in this paper, we introduce a lower bound on the session lifetime in order to minimize the impact of the setup phase on the overall energy consumption. This way, the lower and upper bound of the session duration is defined and a trade-off between the average power consumption and the side-channel resistance can be made.

1.4 Contributions

We summarize the contributions of our work as follows:

- We are the first to perform an in-depth analysis of both the communication and the computation energy cost to achieve end-to-end security and digital signatures for a broad range of IoT platforms, security algorithms and protocols, and wireless communication standards. Three different methods for setting up an end-to-end secured channel and five different modes of operation for the AES algorithm are evaluated.
- We introduce the calculation of a lower bound on the duration of the runtime phase of a security session in order to minimize the overall average power consumption of the session. We evaluate this metric for two different application scenarios, one with a relatively high data rate and one with a relatively low data rate.

1.5 Outline

First, related work is discussed in Sect. 2. Next, the granular approach to evaluate the energy cost of the security computation and communication is described in Sect. 3. Then, the results of the basic operations that are required for the granular approach are presented in Sect. 4. The energy costs of providing an end-to-end secure communication channel, on the one hand, and digital signatures, on the other hand, are evaluated in respectively Sects. 5 and 6. Next, our approach to calculate the minimal session lifetime is presented in Sect. 7. Finally, the paper is concluded in Sect. 8.

2 Related work

Research on the implementation of cryptographic algorithms on MCUs is primarily focused on performance, memory usage, and/or energy consumption [5,6]. For example, the paper published by Ledwaba et al. [5] analyzes the cost of cryptographic software in recent end-point devices. The authors look at the performance of the AES-CTR, SHA256 and ECDSA algorithms. A comparison is made between the different types of ARM Cortex-M devices.

A diverse set of wireless network protocols is analyzed by Sen et al. [7]. The authors evaluate the energy consumption of the network protocols and discuss their security strength.

Although the aforementioned comparisons are useful for practitioners and researchers in the field, they do not give a complete view on the energy requirements of cryptographic protocols providing end-to-end secure communication and digital signatures. Our work performs an in-depth and complete analysis, taking into account the energy of cryptographic computations as well as the energy that is necessary to transmit messages between the IoT device and the end user. Trappe et al. [8] performed a study that is similar to ours. The authors do measurements on the MPS430g2553 16-bit MCU and the CC1150 Multichannel RF Transmitter. However, their work is limited to an experiment that transmits 5 B of data, without taking into account the setup/handshake phase that needs to be executed at the start of each communication session. Other work that considers both the computation and communication energy, is presented by Großschädl et al. [9] and Meulenaer et al. [10]. Both papers compare the energy efficiency of key establishment protocols like Kerberos and ECDH-ECDSA and conclude that the symmetric-key-based Kerberos protocol consumes less energy in almost every scenario. We distinguish ourselves from these existing comparison studies by evaluating the energy consumption of three different implementation platforms, three different wireless communication standards, and several algorithms for both the setup and the runtime phase of a secure communication session. In addition, we analyze digital signatures. We use these results to determine a lower bound on the session duration for two different application scenarios, one in which 33 kB is transmitted every 10 seconds (wearable healthcare device) and one in which 224 B is transmitted every half an hour (weather station).

3 Our approach

The computation and communication energy cost of algorithms and protocols that are required to provide end-to-end security and digital signatures are analyzed using a granular approach. This means that basic operations will be identified for each considered algorithm. The total energy cost of an algorithm is, then, equal to the sum of the energy costs of the required basic operations.

The computation energy cost (E_{comp}) of the basic operations is based on their execution time (t), expressed in number of cycles, in accordance with Eq. (1). The power (P_{comp}) characteristics are determined using the values available in the data sheet of the device, which present the average power consumption per cycle.

$$E_{\text{comp}}(t) = P_{\text{comp}} \times t \tag{1}$$

The communication energy cost ($E_{\text{TX/RX}}$) of the cryptographic protocols is based on the size of the basic messages that need to be communicated. Via the throughput (Th) and power consumption ($P_{\text{TX/RX}}$) of the wireless network, the energy required to send these basic messages of size (M) can be determined via Eq. (2).

$$E_{\text{TX/RX}}(M) = \frac{P_{\text{TX/RX}}}{\text{Th}} \times M \tag{2}$$

Only considering the basic computation and communication elements and the respective featured electrical parameters provides only a rough energy cost estimation. Note that it should provide sufficient accuracy, as the goal is to estimate the relative impact of end-to-end security in terms of communication and computation. Though, other contributing factors like wireless interference and etc. might negatively impact the energy cost of the communication. Thus, we assume that more accurate estimations will have a small impact on the relative share between the computational and the communicational energy cost. Next, the different computation platforms and communication standards are first discussed separately, as a MCU should not be limited to a certain communication standard. After the description, we only consider the combinations that are available according to the used platforms in our results, so that the amount of results are manageable.

3.1 Evaluation platforms

The performance of the basic operations is measured on three different microcontroller platforms. These platforms are chosen to cover a range of different sizes of flash program memory, data memory, and operating frequency. They are described below and more detailed specifications are given in Table 1. The table also indicates the P_{comp} value for each platform, which is necessary for the calculation of the computation energy cost according to Eq. (1).

nuc The NUCLEO-L073RZ is a STM32 Nucleo-64 Development Board of STMicroelectronics [11]. It features the STM32L073RZT6 32 MHz ARM Cortex-M0+ microcontroller with 192 KB flash memory and 20 KB RAM.

msp The TI SimpleLink MSP-EXP432P401R development kit [12] uses the MSP432P401R 48 MHz ARM Cortex-M4F microcontroller with 256 KB flash and 64 KB RAM.

max The MAXREFDES#100 [13] health sensor platform features the MAX32620 96 MHz ARM Cortex-M4F microcontroller with 2 MB flash and 256 KB RAM. It has a wide range of sensors, like a human body temperature sensor and a heart rate sensor.

Table 1 Technical specifications of the considered evaluation platforms

	max	*msp*	*nuc*
MCU	MAX32620	MSP432-P401R	STM32-L073RZT6
Freq. (MHz)	96	48	32
Flash (KB)	2000	256	192
RAM (KB)	256	64	20
Voltage (V)	1.2	3	3
Current (mA)	9.79	7.70	6.65
Power (mW)	11.75	23.10	19.95
- P_{comp}			

3.2 Wireless networks

IoT applications can require a wireless communication range of meters to kilometers. To cover both short-range and long-range applications, we consider the following three protocols:

- Bluetooth low energy (BLE),
- Wi-Fi,
- Long range wide area network (LoRaWAN).

The characteristics of LoRaWAN are provided for two configurations: slowest mode (SF 12) and fastest mode (SF 7). The specifications for all protocols are given in Table 2. All parameters except for the throughput of the BLE network and the energy efficiency are collected from the data sheets of the RF chip that is used to enable the respective communication protocol. Moreover, the values used are primarily from the data sheets' features section. To use the LoRaWAN network, the *nuc* platform is expanded with a Semtech SX1272MB2xAS LoRa extension board [14]. Wi-Fi is provided to the *msp* platform by using the SimpleLink Wi-Fi CC3120 wireless network processor BoosterPack plug-in module. Finally, a BLE compatible RF chip is already present on the *max* platform. It uses the EM9301 BLE controller [15] that supports BLE version 4.1.

The speed at which a wireless network can communicate symbols is not the same as the speed at which data can be sent. The difference is in the overhead of the network protocols in the data link layer. In order to perform an accurate analysis, the actual throughput may have to be calculated. Only the data sheet of the BLE chip does not report on the throughput metric. The LoRa bit rate could also be calculated using the Shannon–Hartley theorem [16]. Additionally, the application needs to take the Fair Policy Access, which limits the amount of data an application is allowed to send [17], into account in a practical LoRa setting. The throughput of the BLE network is calculated using the specifications of the Data link and Physical layer. In BLE version 4.1 [18], a typical message has 14 B of headers and a maximum payload size of 27 B. Furthermore, there is an inter-frame space of 150 μs, i.e., the required interval between two consecutive packets. Taking this into account, the parameters that are necessary to compute the communication energy cost in Eq. (2) are indicated in Table 2.

3.3 Computation

In terms of public-key algorithms, the scope of this paper is limited to those based on elliptic curve cryptography (ECC). These algorithms can be divided into the elliptic curve (EC) arithmetic operations that are used. We consider the following four operations: point addition (PA), point doubling (PD), point multiplication (PM), and fixed-point multiplication (PMG). A distinction is made between a random point multiplication and a fixed-point multiplication, because, optimization techniques like the comb method can be used for the fixed-point multiplication. This optimization requires the pre-calculation of a set of EC points to increase the performance of the multiplication process, but it is only efficient if a point is used multiple times. This is typically the case for the base point (G) of the chosen elliptic curve.

Hash functions and symmetric-key ciphers typically have a much higher performance than public-key-based algorithms. For this reason, these algorithms were not divided into basic operations but are considered basic operations themselves. The following hash functions are considered: SHA256 and SHA3-256. For the symmetric-key ciphers, AES is used in the following five modes of operation: electronic codebook (ECB), cipher block chaining (CBC), counter (CTR), counter with CBC-MAC (CCM), and Galois/counter mode (GCM). ECB is the most naive form of encryption and should never be used on its own, but, it provides a benchmark of the most basic form of AES encryption. The CBC mode can be used as a primitive in either encryption or integrity protection algorithms. Next, CTR mode is typically used to provide confidentiality. The last two cipher modes can be categorized as authenticated encryption with associated data (AEAD). They provide both confidentiality and data authenticity (integrity).

The performance of all identified basic operations is measured on the three platforms. Fifty-time measurements are done for each basic operation using the platforms' available timer. Moreover, the AES cipher operation is an encryption on 256 Bytes of data. We have chosen a multiple of the AES block size, because longer time periods ensure less influence of potential timing inaccuracies like an early start and late end. For the hash function, the maximum input size of the respective algorithm for one round is chosen as follows: 55 B for SHA256 and 135 B for SHA3-256. The total available internal state size is not used for SHA256 and SHA3-256,

Table 2 Technical specifications of the considered wireless protocols

	BLE (4.1)	Wi-Fi	LoRA (SF 12)	LoRA (SF 7)
Electrical characteristics				
Evaluation platform	*max*	*msp*	*nuc*	*nuc*
RF chip	EM9301	CC3120	SX1272	SX1272
Bandwidth (MHz)	1	20	0.125	0.25
Symbol rate (kbps)	1000	54000	0.366	13.7
Throughput (kbps)—Th	305	16000	0.293	10.94
Range (km)	0.1	0.25	14	2
Voltage (V)	2.5	3.6	3.3	3.3
TX				
Current (mA)	12	59	28	28
Power (mW)—P_{TX}	30	212.4	92.4	92.4
Energy efficiency ($\mu J/B$) $-8 \times P_{TX}/Th$	**0.79**	**0.11**	**2523**	**67.58**
RX				
Current (mA)	13	229	11.2	11.2
Power (mW)—P_{RX}	32.5	824	37.0	37.0
Energy efficiency ($\mu J/B$) $-8 \times P_{RX}/Th$	**0.852**	**0.412**	**1009.3**	**27.034**
Idle				
Current (mA)	0.009	0.690	1.4	1.4
Power (mW)	0.023	2.484	4.62	4.62

The energy consumption for transmitting and receiving messages in bold

as we take into account the minimal padding or suffix that is required for the last block of input data. Note that the most optimal scenario, i.e., the maximum amount of input data to fill up the internal state completely, is used for each of the operations. Bar charts with error bars and the 95% percentile are used to represent the results and the spread. The energy cost is calculated using Eq. (1). Additionally, the energy results of the AES ciphers and hash algorithms are divided by their message input size to calculate the energy required per byte.

All basic operations are implemented using software libraries and cross-compiled with the GNU Tools for ARM Embedded Processors version 6-2017-q2-update. Furthermore, the compiler is configured to optimize for size (-Os) to limit the storage requirement of the implementations, as public-key-based algorithms typically take up a lot of the already limited available storage of constrained embedded devices. We have chosen to utilize the default configuration for the different cryptographic implementations which typically provide generic optimizations that apply to all types of constrained platforms. However, certain optimizations provide additional benefits in terms of performance and are consequently more energy efficient. Nevertheless, the focus of this work is not on the optimization of cryptographic implementations but rather on the comparison of the energy efficiency.

The RELIC-toolkit library [19] is used to implement the EC arithmetic and the SHA256 hash function. We use the SECG K-256 prime elliptic curve, "BASIC;COMBA;COMBA;-MONTY;MONTY;SLIDE" configuration for the prime field arithmetic, and "PROJC;LWNAF;COMBS;INTER" configuration for the prime elliptic curve arithmetic. For more information on how to configure RELIC and other examples that use it, we refer to the wiki [19] and to an example [20]. The AES ciphers are implemented using Mbed TLS [21] and SHA3 using wolfCrypt [22]. We use the SHA3-256 hash function as specified in FIPS PUB 202 [23].

The performance measurements have been published on the Zenodo platform under the Creative Commons Attribution 4.0 International license [24]. Furthermore, the results of the ECC arithmetic, AES block cipher, and hash algorithms benchmark are visualized in respectively Figs. 1, 2 and 3. If we analyze the differences between the algorithms, the performance results of the public-key algorithms are higher by a factor of 1000 compared to the symmetric-key algorithms, with the point multiplication operation requiring the most computation time. Also, it is noticeable that the AES operations without authenticated encryption outperform the hash algorithms, e.g. an AES ECB encryption needs about 10% up to 44% less computation time than a SHA256 calculation. On the other hand, the authenticated encryption ciphers require more computation time than the hash algorithms. For exam-

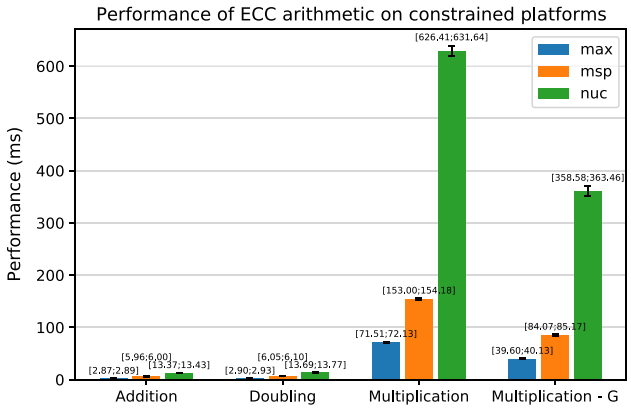

Fig. 1 Performance results of elliptic curve arithmetic on the three considered platforms (PA: point addition, PD: point doubling, PM: point multiplication, PMG: fixed-Point multiplication)

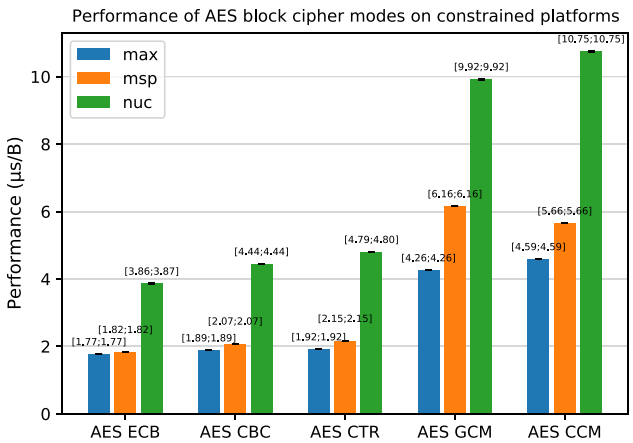

Fig. 2 Performance results per byte of AES with modes of operation on the three considered platforms

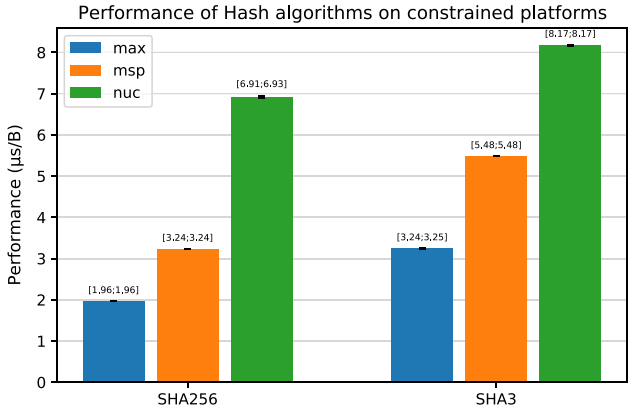

Fig. 3 Performance results per byte of the SHA256 and SHA3-256 hash functions on the three considered platforms

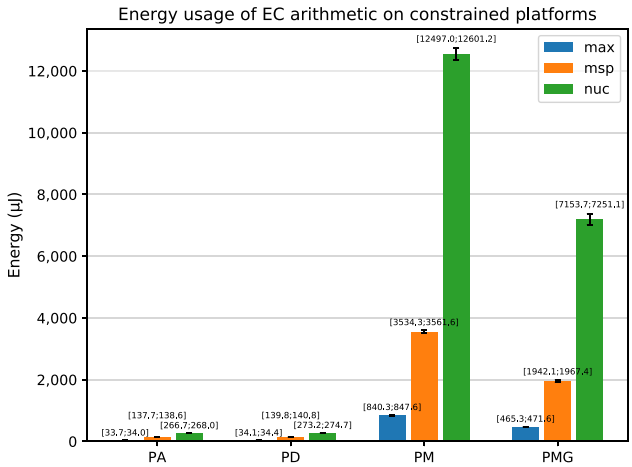

Fig. 4 Energy cost of elliptic curve arithmetic on the three considered platforms (PA: point addition, PD: point doubling, PM: point multiplication, PMG: fixed-Point multiplication)

ple, an AES GCM encryption requires around 117% down to 44% more computation time than a SHA256 calculation. Finally, the performance results show that SHA256 is 15% up to 40% more efficient than SHA3, though SHA256 has a smaller maximum message input size per round of 55 bytes compared to the 135 bytes of SHA3.

3.4 Communication

Public-key-based algorithms typically need to communicate a selection of the following four basic objects: an EC point, a signature, a random number, and/or a certificate. The curve we use throughout the paper is SECG K-256. For this curve, an uncompressed point can be compiled in 65 Bytes using the SEC1 encoding. The size of a signature depends on the algorithm used, e.g., the ECDSA scheme produces a 64 Byte signature. A typical random number uses 32 Bytes. Finally, the size of a certificate depends on the type. For example, a SECG K-256-based X.509 certificate requires 544 Bytes. This value was measured by generating a basic certificate using the OpenSSL command line interface.

4 Energy results for the basic operations

The energy cost of the elliptic curve arithmetic is presented in Fig. 4. The use of the comb method for the fixed point multiplication explains the enhanced performance in comparison to the general point multiplication. Furthermore, the addition and doubling operations pale in comparison with the point multiplication.

The results of the SHA256 and SHA3-256 hash function are presented in Fig. 5. Note the different scale on the vertical axis. Similar as shown in the performance results, SHA256 is about 40% more energy efficient than SHA3 in terms of energy per byte. In terms of performance, one round of SHA3 is much slower than one round of SHA256, but, it can input more bytes per round. SHA256 can input 55 B while SHA3

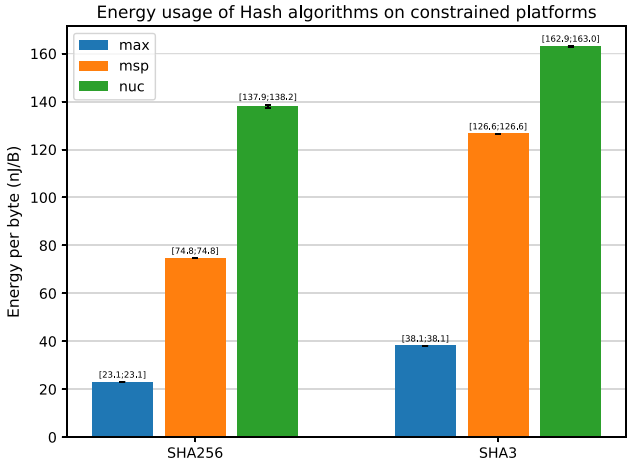

Fig. 5 Energy cost per byte of the SHA256 and SHA3-256 hash functions on the three considered platforms

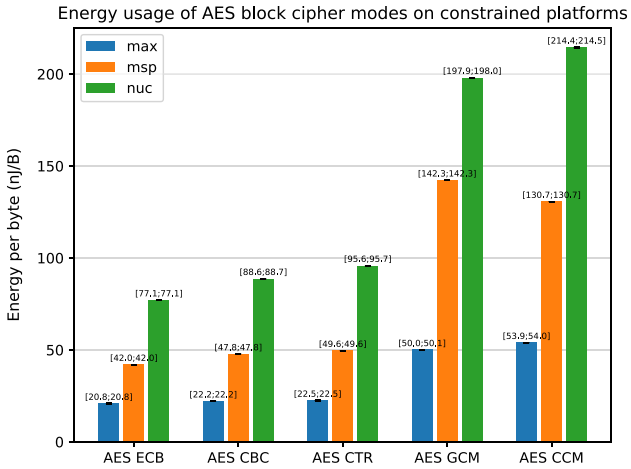

Fig. 6 Energy cost per byte of AES with modes of operation on the three considered platforms

can handle 135 B of data. On the max platform, one round takes around 108 μs and 438 μs for respectively SHA256 and SHA3. The advantage of SHA3 is the strong security rationale behind it. We refer to the Keccak reference [25] for more information.

The results of the AES cipher and operation modes are shown in Fig. 6. The simplest block cipher mode, ECB, is, as expected, the most efficient with about 21 nJ/B and 77 nJ/B for respectively the max and nuc platform. The CBC and CTR mode express similar but slightly higher energy costs. The AEAD cipher modes, GCM and CCM, require about double the energy per byte in comparison to the basic modes. This can be attributed to the authentication operation that is additionally provided.

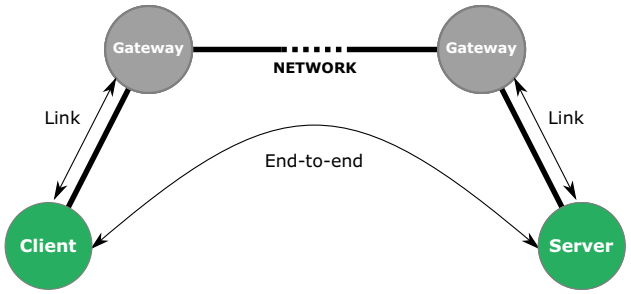

Fig. 7 Schematic representation of an end-to-end secured connection

5 Energy results for end-to-end security

A secure communication channel is the channel used between two parties (client and server) to provide end-to-end security guarantees as shown in Fig. 7. It is used to provide confidentiality, integrity, and mutual authenticity. Throughout this section, the constrained device is used as the server and a remote party as a client. Two methods of providing security are considered in this paper: via encryption and via signatures. The use of encryption implies the use of a symmetric-key algorithm like AES, and is typically much more efficient than the use of a public-key algorithm. However, it relies on having a shared encryption key. Public-key algorithms can provide data authentication by generating signatures on messages. They have the advantage of having a set of two keys: a private key and a public key. One key can be used to sign a message and the other one to verify the signature, and thus, a shared key is not required. Furthermore, one set of keys is typically linked to an entity via, e.g., a certificate. This ensures that entities can be authenticated.

For applications that use encryption for end-to-end security, three methods of establishing a shared key are analyzed in this chapter: pre-shared key (PSK), pre-shared public key (PSPK), and pre-shared trusted third party public key (PSTTPPK). PSK implies that a shared encryption key is already present on both communicating entities. This is typically done via an out-of-band method. Authentication of the entities is, then, implicitly assumed through the use of this shared key. PSPK means that the public key of the remote party is pre-configured, which is also done via an out-of-band method. This ensures that the authentication does not rely on a shared secret but on the use of the public key and the corresponding private key. However, both the PSK and the PSPK methods have one important drawback. The keys are pre-configured, and therefore, the keys must be managed internally. An update may be required due to, e.g., the shared key being compromised. The management of these keys must be done manually or, e.g., using a custom implementation on a local server. For this reason, a trusted third party (TTP) is typically used. When using PSTTPPK, the TTP generates, for example, a certificate to link a public key to an entity.

Then, other entities can, via the public key of the TTP, verify the certificate and thus the link between the public key and the entity. Thus, as long as the TTP is not compromised, entities can verify the authenticity of one another.

Securing a channel via encryption can typically be divided into two phases: a setup phase (session setup), and a runtime phase (session runtime). For example, the TLS protocol version 1.2 [26] uses a similar approach. During the session setup phase, a session encryption key is generated and both communicating parties are authenticated. This session key is, then, used by the symmetric-key algorithm to encrypt the communication between the two entities.

5.1 Session setup

The PSK method uses two random values padded to the PSK as input to a key derivation function (KDF) to generate a session encryption key. The PSK is not used as a session key such that the system cannot be compromised by attacking the session key. First, both the client and server generate and communicate to each other their random value. The random value is generated using the Hash-based deterministic random bit generator (Hash-DRBG). It is also a source of randomness recommended by NIST [27]. Then, a session encryption key is derived. The Hash-based key derivation function (HKDF) is used as KDF.

Another method to generate the session encryption key is to use public-key-based cryptography like the Diffie–Hellman (DH) technique. The DH scheme can create a shared secret over an unsecured channel. In the PSPK method, the Ephemeral Elliptic Curve Diffie–Hellman (ECDHE) key agreement protocol is used. In short, both entities generate a new set of public and private keys (ephemeral keys) and send each other the public key. Via the DH scheme, a shared secret is generated. To ensure the entity authentication of both the client and server, it is assumed that the ephemeral public keys are signed using the entity's respective public key. Both parties now have a shared secret, but it is not used as the session encryption key for the same reason as in the PSK method. Thus, random numbers need to be generated and exchanged, and, a session key is derived using the key derivation function. The Hash-DRBG and HKDF algorithm are also used for this purpose in this method.

The difference of the PSTTPPK method with respect to the PSPK method is that the public keys of the communicating entities are exchanged to each other and validated using the TTP's public key. Thus, during the setup phase, both entities send each other their certificate. This certificate is signed by the TTP. In this work, the use of X.509 certificates is assumed. Then, each entity validates the certificate by checking the signature. The remainder of the setup phase is the same as the setup of the PSPK method.

Table 3 The computation cost of the considered algorithms divided into basic operations (H: hash function on a Message (M), PMG: fixed-point multiplication, PM: point multiplication, and PA: point addition)

Algorithm	Computation cost
Hash DRBG (R)	$4\|H(M)\|$
HKDF (KDF)	$8\|H(M)\|$
ECDSA—sign	$\|H(M)\| + \|R\| + \|PMG\|$
ECDSA—verify	$\|H(M)\| + \|PMG\| + \|PM\| + \|PA\|$
ECDHE	$\|R\| + \|PMG\| + \|PM\| + \|KDF\|$

Table 4 The computation cost of the considered session setup methods (R: Hash DRBG, KDF: HKDF, DHE: ECDHE, and sign/verify: ECDSA)

Sess. setup	Computation cost
PSK	|R| + |KDF|
PSPK	|R| + |DHE| + |sign| + |verify| + |KDF|
PSTTPPK	|R| + |DHE| + |sign| + 2|verify| + |KDF|

Table 5 The communication cost of the considered session setup methods (C: certificate, R: random value, Y: elliptic curve point, and S: signature) and the number of transmitted bytes

Session setup	Communication	Bytes
Input		
PSK	$\|R\|$	32
PSPK	$\|R\| + \|Y\| + \|S\|$	161
PSTTPPK	$\|R\| + \|C\| + \|Y\| + \|S\|$	705
Output		
PSK	$\|R\|$	32
PSPK	$\|R\| + \|Y\| + \|S\|$	161
PSTTPPK	$\|R\| + \|C\| + \|Y\| + \|S\|$	705

The granular approach to estimate the computation energy cost of the previously mentioned algorithms is presented in Table 3. Only the most computation-intensive operations are considered, similar to the approach of Saeed et al. [28]. Furthermore, specific optimizations to these algorithms, like the Shamir's trick used in ECDSA, are not factored in. Next, the computation energy cost of the three session setup methods is presented in Table 4.

The communication energy cost is estimated for the three session setup methods and the results are presented in Table 5. Only the communication of basic objects is assumed. This is in contrast to the TLS protocol, where a lot of overhead messages are required to first agree upon a cipher suite etc.

The total energy cost of the three session setup methods are calculated using the granular approach. First, the communication and computation energy cost are separately presented in Fig. 8. Next, three combinations of the tested MCUs and

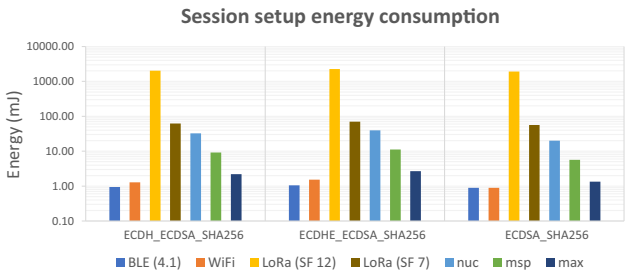

Fig. 8 Energy consumption of the three configurations for the wireless networks and MCUs separately

Table 6 The total energy consumption and the relative share of the computation and communication energy cost for the three considered session setup methods and the three MCU and platform combinations

Configuration	PSK	PSPK	PSTTPPK
nuc + LoRa (SF7)			
Total (μJ)	3119	62345	133845
Comp (%)	2.9%	75.6%	50.2%
Comm (%)	97.1%	24.4%	49.8%
msp +WiFi			
Total (μJ)	66	13257	19185
Comp (%)	74.8%	99.4%	98.1%
Comm (%)	25.2%	0.6%	1.9%
max + BLE (4.1)			
Total (μJ)	68	3414	5654
Comp (%)	22.3%	92.3%	79.6%
Comm (%)	77.7%	7.7%	20.4%

wireless networks are made: nuc + LoRa (SF7), msp + Wi-Fi, and max + BLE (4.1). Only the best-case scenario for LoRa (SF7) is taken into account. The total energy consumption and the relative share of the computation and the communication are presented in Table 6.

The PSK method requires the least amount of energy, and, the communication energy cost outweighs the computation cost for the nuc and max platforms, but not for the msp platform. The reason is that Wi-Fi has a higher energy efficiency than BLE and LoRa. For the PSPK and PSTTPPK method, the computation energy cost has a much larger impact than the communication energy cost. However, the LoRa network, which is more energy efficient, suffers more from sending e.g. the certificate. For the PSTTPPK method on the nuc platform, the energy cost is almost equally divided over the computation and communication energy cost. Note that the results are a rough estimation and they are probably lower than the actual energy consumption. We have only taken into account the best case scenario e.g. perfect wireless conditions, no header overhead, etc. However, these rough estimations are sufficient to show trends and compare different approaches.

5.2 Session runtime

The AES algorithm is used to provide the encryption of data. In terms of computation cost, the AES block cipher modes are considered as basic operations. In terms of communication cost, only the AEAD block ciphers require additional data besides the message to be sent. Thus, an authentication tag of 12 B is taken into account. The total energy consumption and the impact of the computation and communication are presented in Table 7.

The communication requires significantly more energy than the computation. For example, the computation only takes about 2–6% of the total energy cost for the max platform. However, the total energy cost is almost spread evenly for the msp platform. The computation uses about 9–32% of the total energy cost for AES ECB, CTR and CBC, while it takes about 24–57% for the AEAD ciphers. For the nuc platform that uses the LoRa network, the computation cost is almost negligible. It requires around 0.1–0.8% of the total energy cost. Note that the Wi-Fi wireless communication has the best theoretical energy efficiency of the considered wireless communication standards, as indicated in Table 2. Besides providing a rough and best case estimation, Wi-Fi has a much higher theoretical data throughput.

6 Energy results for digital signatures

Another method of securing data is through the use of signatures. The following two algorithms are considered: the Schnorr signature scheme and the Signcryption scheme. The Schnorr signature scheme is a public-key-based algorithm that relies on the public key of the sender to generate a signature on a message. This signature can be used to verify the authenticity of the message. Next, the Signcryption scheme is similar to the Schnorr scheme but provides besides the authenticity also the confidentiality guarantee. Simply put, the public-key operations generate besides a signature also a secret key. This secret key can only be derived by the sender and intended receiver of the message. Using this key, a symmetric-key-based algorithm can encrypt the message and provide confidentiality. It is assumed that AES-CTR is used in the Signcryption scheme.

The computation energy cost is again estimated using the granular approach. The required basic operations are listed in Table 8. In terms of communication, both schemes produce a signature of 64 B. Also, we assume that the public keys of both parties are pre-configured and do not require additional communication or verification.

The energy efficiency in function of the data packet size is presented in Fig. 9. It decreases with increasing message size, because, the public-key-based operations only need to be performed once per message. As a reference, the energy

Table 7 The total energy consumption and the relative share of the computation and communication energy cost for the considered session runtime algorithms and MCU and platform combinations. (LoRa: SF7, BLE: v4.1)

Algorithm	TX (AES)					RX (AES)				
	ECB	CBC	CTR	GCM	CCM	ECB	CBC	CTR	GCM	CCM
nuc+LoRa										
Total ($\mu J/B$)	67.66	67.67	67.68	67.78	67.80	27.11	27.12	27.13	27.23	27.25
Comp (%)	0.1%	0.1%	0.1%	0.3%	0.3%	0.3%	0.3%	0.4%	0.7%	0.8%
Comm (%)	99.9%	99.9%	99.9%	99.7%	99.7%	99.7%	99.7%	99.6%	99.3%	99.2%
msp+WiFi										
Total ($\mu J/B$)	0.15	0.15	0.16	0.25	0.24	0.45	0.46	0.46	0.56	0.54
Comp (%)	28.4%	31.0%	31.8%	57.3%	55.2%	9.3%	10.4%	10.7%	25.7%	24.1%
Comm (%)	71.6%	69.0%	68.2%	42.7%	44.8%	90.7%	89.6%	89.3%	74.3%	75.9%
max+BLE										
Total ($\mu J/B$)	0.81	0.81	0.81	0.84	0.84	0.87	0.88	0.88	0.90	0.91
Comp (%)	2.6%	2.7%	2.8%	6.0%	6.4%	2.4%	2.5%	2.6%	5.5%	6.0%
Comm (%)	97.4%	97.3%	97.2%	94.0%	93.6%	97.6%	97.5%	97.4%	94.5%	94.0%

Table 8 The computation cost of the considered public-key-based algorithms divided into basic operations (H: Hash function on a Message (M), Y: elliptic curve point, PMG: Fixed-Point Multiplication, PM: Point Multiplication, PA: Point Addition, E: Encryption, and D: Decryption)

Algorithm	Computation cost
Schnorr	
Sign	$\|R\| + \|PMG\| + \|H(M+Y)\|$
Verify	$\|PM\| + \|PMG\| + \|PA\| + \|H(M+Y)\|$
Signcryption	
Sign	$\|R\| + \|PMG\| + \|PM\| + \|H(M+3Y)\| + \|E(M)\|$
Verify	$2\|PM\| + \|PMG\| + \|PA\| + \|H(M+3Y)\| + \|D(M)\|$

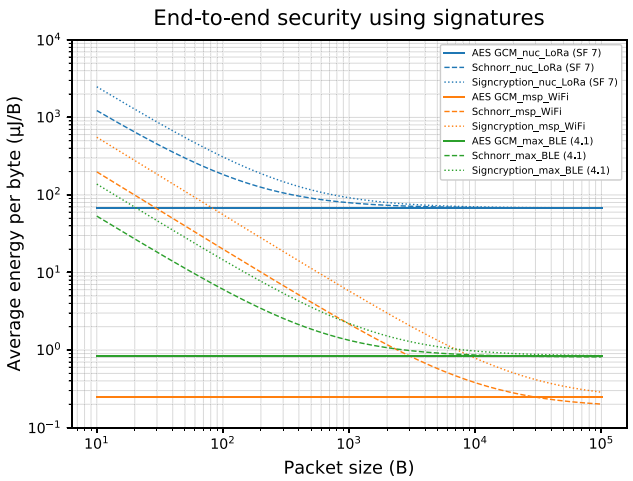

Fig. 9 The energy efficiency of the Schnorr signature scheme and signcryption scheme as a function of the message size for each MCU and wireless network combination. The energy efficiency of AES-GCM is also added as a reference value

efficiency of AES-GCM is added to the graph. In terms of efficiency, it requires a message size of about 5–50 kB to reach the efficiency of AES-GCM. However, we do not take the key establishment requirement of symmetric-key based end-to-end security into account. Furthermore, the coupon technique could be used to omit the required public-key-based operations of both the Schnorr and signcryption scheme as explored by Winderickx et al. [29].

7 Derivation of the minimal session period

Section 5.1 shows that the computation and communication energy impact of the session setup phase is considerable. However, the energy consumption impact can be minimized by looking at the big picture of providing a secured communication channel, i.e., taking into account that a session can be divided into a setup phase and a runtime phase. The impact can be reduced by controlling the frequency at which the session setup phase is executed. First, an equation for the minimum time period of a session is derived. The equation is then applied to two IoT applications: a wearable healthcare device and a weather station.

The total average power consumption P_{total} of a session is defined by the average power consumption of the session runtime P_{runtime} and the energy cost of the session setup E_{setup} divided by the time period of the session T, see Eq. 3.

$$P_{\text{total}}(T) = \frac{E_{\text{setup}}}{T} + P_{\text{runtime}} \tag{3}$$

The average runtime power consumption (P_{runtime}) can be estimated using Eq. (4). In a practical scenario, it can also be measured. In this equation, the transmission rate of the message, the total amount of application data in a message and the amount of additional data are denoted by, respectively, R_{app}, N_{app} and N_{add}. The data additionally generated by the encryption, e.g., the MAC, is counted as additional data. Note that it is assumed that the application only transmits data, and that the platform's and wireless network's idle energy consumption are negligible. In the equation, the amount of data is multiplied by the message transmission rate and the respective energy cost per byte for the communication (E_{TX}) and the computation (E_{AES}) aspect.

$$P_{\text{runtime}} = (N_{\text{app}} + N_{\text{add}})R_{\text{app}}E_{\text{TX}} + N_{\text{app}}R_{\text{app}}E_{\text{AES}} \tag{4}$$

The energy cost of the session setup is estimated using the results of Sect. 5. In a practical setting, a separate benchmark of the entire setup phase of, e.g., the TLS protocol could be done to produce more accurate results.

As an example, we could state that the average power consumption impact of the session setup phase should be limited to 5% of the session runtime's average power consumption. This would then lead to Eq. (5).

$$P_{\text{setup}} = 0.05\,P_{\text{runtime}} \tag{5}$$

Since $P_{\text{setup}} = E_{\text{setup}}/T$, the equation to calculate the minimal time period of the session can be derived, see Eq. (6).

$$T_{\text{minimal}} = \frac{E_{\text{setup}}}{0.05\,P_{\text{runtime}}} \tag{6}$$

The total average power consumption as a function of the time period of a session is plotted in Figs. 10 and 11 for the healthcare and the weather station scenario, respectively. The wearable healthcare device transmits 33,000 kB of data every 10 s [30]. The weather station transmits about 224 B of data every half hour [31]. The following three configurations of the session setup and runtime phase are considered: PSK + AES-GCM, PSPK + AES-GCM, and PSTTPPK + AES-GCM. Additionally, the minimal time period based on the 5% rule that we introduced as an example is calculated for all configurations.

The healthcare scenario is plotted in Fig. 10. The nuc and LoRa combination clearly does not fit this setting, because it takes too much energy to transmit the high amount of data. The best fit is the Wi-Fi network combined with the msp platform for all configurations. Furthermore, the msp platform has better results than the max platform, because, the Wi-Fi network chip features better energy efficiency. The minimal session period based on the 5% rule ranges from 10.1 to 330.2 s. It is very low, because the healthcare scenario requires a lot of energy to send its application data.

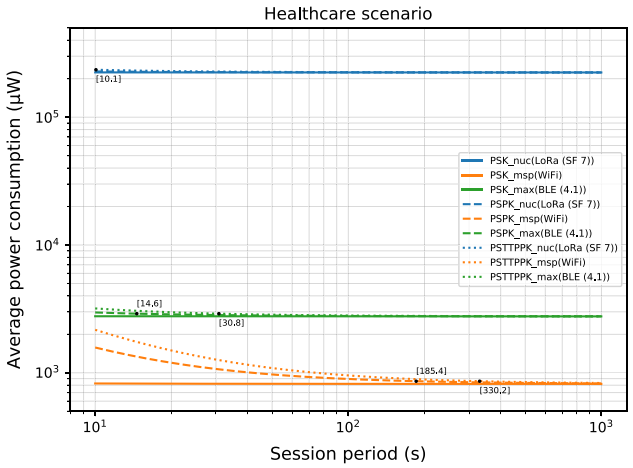

Fig. 10 Average power consumption as a function of the session's time period for the healthcare scenario, where the minimal time period based on the 5% rule is plotted using a black dot

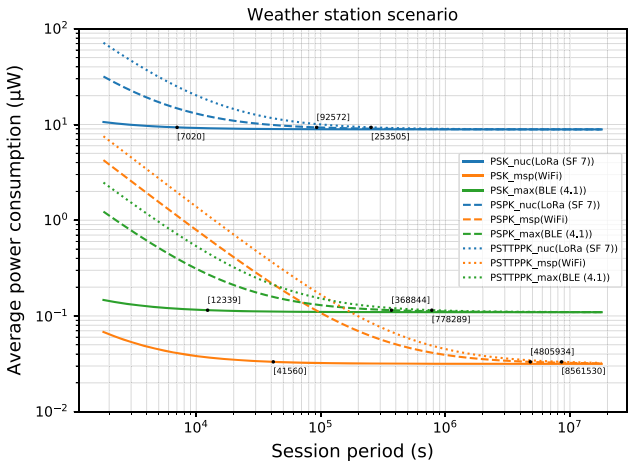

Fig. 11 Average power consumption as a function of the session's time period for the weather station scenario, where the minimal session period based on the 5% rule is plotted using a black dot

Furthermore, it should be noted that this time period is a lower bound.

The weather station scenario is shown in Fig. 11. Depending on the position and range requirements of the application, all combinations are possible. The Wi-Fi network can provide the lowest average power consumption. The LoRa network consumes the most energy, but it has the advantage of range over the BLE and Wi-Fi network. In this lower data throughput scenario, the minimal session period based on the 5% rule is considerably higher than in the healthcare scenario. It ranges from 1.9 h to 99 days.

8 Conclusion

A granular approach was used to estimate the computation and communication energy cost of algorithms used to provide end-to-end security on the one hand and digital signatures on the other hand. Measurements were done on three platforms (nuc, msp and max) using three different wireless communication protocols (BLE, Wi-Fi and LoRaWAN). First, basic operations of the considered algorithms were identified and benchmarked. Then, the computation energy cost of these basic operations were evaluated. To explore the impact of both the computation and communication energy cost, two security techniques were explored: end-to-end encryption and digital signatures. For end-to-end encryption, both the session setup phase and the session runtime phase were taken into account. For the session setup, the energy results showed that the computation energy cost outweighed the communication cost in most evaluated scenarios. For the session runtime phase, this was the other way around. The results of the purely signature-based security turned out to be not suitable for IoT applications without additional computation optimization. The Schnorr and signcryption scheme reached the energy efficiency level of AES-GCM at about 5–50 kB of processed and transmitted data.

We also introduced an equation to calculate a recommended lower bound on the lifetime of a session to enable the developer to find a trade-off between the average power consumption and the side-channel resistance. In terms of security, the highest security level is achieved when the session is continuously renewed. On the other hand, the lowest average power consumption is obtained when the session is never renewed. In the equation, the developer can define the relative share of the application's total energy budget that may be used for renewing the security session. In our examples, we have chosen to dedicate 5% of the total energy budget to the security sessions. The equation was applied to the following two IoT scenarios: a healthcare and a weather station application. The relative energy share of the session setup phase in the overall energy consumption was almost negligible, i.e. lower than 5%, for minimal time periods in the range of 30–330 s for the different IoT platforms in the healthcare application. For the weather station application, the minimal time period for making the session setup energy negligible ranged from about 1.9 h to 99 days for the different IoT platforms. When the minimal time period is taken into account, both the symmetric-key and public-key-based key establishment protocols can be suitable for IoT applications.

In summary, this paper gave an overview of the energy impact of different security schemes for the IoT, taking into account both the computation and the communication energy. It also used this information to determine the minimal session period needed to make the session setup energy negligible. The paper serves as a guideline for practitioners and researchers selecting the appropriate security algorithms and wireless communication protocols, and determining the minimal session period in order to minimize the overall energy consumption.

References

1. Sornin, N., Luis, M., Eirich, T., Kramp, T., Hersent, O.: LoRaWAN Specification V1.0. Tech. rep., LoRa Alliance (2015)
2. Kocher, P., Jaffe, J., Jun, B.: Differential power analysis. In: Wiener, M. (ed.) Advances in Cryptology: Proceedings of CRYPTO'99, no. 1666 in Lecture Notes in Computer Science, pp. 388–397. Springer, Berlin (1999)
3. Gandolfi, K., Mourtel, C., Olivier, F.: Electromagnetic analysis: concrete results. In: Koç, Ç.K., Naccache, D., Paar, C. (eds.) Proceedings of 3rd International Workshop on Cryptographic Hardware and Embedded Systems (CHES), no. 2162 in Lecture Notes in Computer Science, pp. 255–265. Springer, Berlin (2001)
4. Kocher, P.: Timing attacks on implementations of Diffie-Hellman, RSA, DSS, and other systems. In: Annual International Cryptology Conference, pp. 104–113. Springer, Berlin (1996)
5. Ledwaba, L.P.I., Hancke, G.P., Venter, H.S., Isaac, S.J.: Performance costs of software cryptography in securing new-generation internet of energy endpoint devices. IEEE Access **6**, 9303 (2018). https://doi.org/10.1109/ACCESS.2018.2793301
6. de Clercq, R., Uhsadel, L., Van Herrewege, A., Verbauwhede, I.: Proceedings of the the 51st annual design automation conference on design automation conference—DAC '14. In: Proceedings of the the 51st Annual Design Automation Conference on Design Automation Conference—DAC '14, pp. 1–6. ACM Press, New York, New York, USA (2014). https://doi.org/10.1145/2593069.2593238. http://dl.acm.org/citation.cfm?doid=2593069.2593238
7. Sen, S., Koo, J., Bagchi, S.: TRIFECTA: security, energy efficiency, and communication capacity comparison for wireless IoT devices. IEEE Internet Comput. **22**(1), 74 (2018). https://doi.org/10.1109/MIC.2018.011581520
8. Trappe, W., Howard, R., Moore, R.S.: Low-energy security: limits and opportunities in the Internet of Things. IEEE Secur. Priv. **13**(1), 14 (2015). https://doi.org/10.1109/MSP.2015.7
9. Großschädl, J., Szekely, A., Tillich, S.: The energy cost of cryptographic key establishment in wireless sensor networks. In: Proceedings of the 2nd ACM Symposium on Information, Computer and Communications Security, ASIACCS '07, pp. 380–382. Association for Computing Machinery, New York, NY, USA (2007). https://doi.org/10.1145/1229285.1229334
10. de Meulenaer, G., Gosset, F., Standaert, F., Pereira, O.: On the energy cost of communication and cryptography in wireless sensor networks. In: 2008 IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, pp. 580–585 (2008). https://doi.org/10.1109/WiMob.2008.16
11. STMicroelectronics. NUCLEO-L073RZ (2019). https://www.st.com/en/evaluation-tools/nucleo-l073rz.html. [Online; accessed June-2019]
12. Texas Instruments Incorporated. Simplelink^{TM} msp432p401r high-precision adc launchpad^{TM} development kit (2019). http://www.ti.com/tool/MSP-EXP432P401R. [Online; accessed June-2019]
13. maxim integrated. MAXREFDES100#: Health Sensor Platform (2019). https://www.maximintegrated.com/en/design/reference-design-center/system-board/6312.html. [Online; accessed June-2019]
14. Semtech Corporation. SX1272/73—860 MHz to 1020 MHz Low Power Long Range Transceiver. https://www.semtech.

com/uploads/documents/SX1272_DS_V4.pdf (2019). [Online; accessed June-2019]
15. em microelectronics. EM9301—Single-Cell Battery Bluetooth Low Energy Controller (2018). https://www.emmicroelectronic.com/product/standard-protocols/em9301. [Online; accessed June-2019]
16. Cheong, P.S., Bergs, J., Hawinkel, C., Famaey, J.: Comparison of LoRaWAN classes and their power consumption. In: 2017 IEEE Symposium on Communications and Vehicular Technology (SCVT), vol. 2017-Decem, pp. 1–6. IEEE, Heverlee, Belgium (2017). https://doi.org/10.1109/SCVT.2017.8240313. http://ieeexplore.ieee.org/document/8240313/
17. The Things Network. Duty Cycle for LoRaWAN Devices (2019). https://www.thethingsnetwork.org/docs/lorawan/duty-cycle.html. [Online; accessed June-2019]
18. Core System Package [Low Energy Controller volume] Bluetooth, SIG, Specification of the Bluetooth System v4.1 **6**, 2467 (2013)
19. Aranha, D.F., Gouvêa, C.P.L.: RELIC is an efficient library for cryptography. https://github.com/relic-toolkit/relic (2009)
20. Hummen, R., Ziegeldorf, J.H., Shafagh, H., Raza, S., Wehrle, K.: Towards viable certificate-based authentication for the internet of things. In: Proceedings of the 2nd ACM workshop on Hot topics on wireless network security and privacy—HotWiSec '13, p. 37. ACM Press, New York, New York, USA (2013) https://doi.org/10.1145/2463183.2463193. http://dl.acm.org/citation.cfm?doid=2463183.2463193
21. ARM Holdings. Arm Mbed TLS (2019). https://tls.mbed.org/. [Online; accessed June-2019]
22. WolfSSL. wolfCrypt Embedded Crypto Engine (2019). https://www.wolfssl.com/products/wolfcrypt-2/
23. Dworkin, M.J.: SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD (2015). https://doi.org/10.6028/NIST.FIPS.202
24. Winderickx, J., Braeken, A., Singelée, D., Mentens, N.: Performance measurements for in-depth energy analysis of security algorithms and protocols for the internet of things. Dataset on Zenodo. https://doi.org/10.5281/zenodo.3957700
25. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: The Keccak reference. Tech. rep, STMicroelectronics, NXP Semiconductors (2011)
26. Rescorla, E., Dierks, T.: The Transport Layer Security (TLS) Protocol Version 1.2. RFC 5246 (2008). https://doi.org/10.17487/RFC5246. https://rfc-editor.org/rfc/rfc5246.txt
27. Barker, E.B., Kelsey, J.M.: Recommendation for Random Number Generation Using Deterministic Random Bit Generators. Tech. rep., National Institute of Standards and Technology, Gaithersburg, MD (2015). https://doi.org/10.6028/NIST.SP.800-90Ar1. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf
28. Saeed, M.E.S., Liu, Q.Y., Tian, G., Gao, B., Li, F.: AKAIoTs: authenticated key agreement for Internet of Things. Wirel. Netw. **25**(6), 3081 (2019). https://doi.org/10.1007/s11276-018-1704-5
29. Winderickx, J., Braeken, A., Mentens, N.: Storage and computation optimization of public-key schemes on embedded devices. In: 2018 4th International Conference on Cloud Computing Technologies and Applications (Cloudtech), pp. 1–8. IEEE, Brussel (2018). https://doi.org/10.1109/CloudTech.2018.8713334. https://ieeexplore.ieee.org/document/8713334/
30. Winderickx, J., Bellier, P., Duflot, P., Coppieters, D., Mentens, N.: WiP: communication and security trade-offs for wearable medical sensor systems in hospitals. In: Proceedings of the ACM SIGBED International Conference on Embedded Software (EMSOFT), p. 2. ACM, New York, NY, USA (2019)
31. Roussey, C., Bernard, S., Andre, G., Corcho, O., Sousa, G.D., Boffety, D., Chanet, J.P.: Short paper: weather station data publication at IRSTEA: an implementation report. In: Kyzirakos, K., Gruetter, R., Kolas, D., Perry, M., Compton, M., Janowicz, K., Taylor, K. (eds.) Joint Proceedings of the 6th International Workshop on the Foundations, Technologies and Applications of the Geospatial Web and 7th International Workshop on Semantic Sensor Networks (TC-SSN) (Aachen, 2014), no. 1401 in CEUR Workshop Proceedings, pp. 89–104. http://ceur-ws.org/Vol-1401/#paper-07