Computer Networks 102 (2016) 83–95

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Survey paper

On perspective of security and privacy-preserving solutions in the internet of things

Lukas Malina*, Jan Hajny, Radek Fujdiak, Jiri Hosek

Department of Telecommunications, Brno University of Technology, Brno, Czech Republic

ARTICLE INFO

Article history:
Received 14 July 2015
Revised 15 January 2016
Accepted 23 March 2016
Available online 26 March 2016

Keywords:
Cryptography
Internet of things
Performance evaluation
Privacy
Security

ABSTRACT

The Internet of Things (IoT) brings together a large variety of devices of different platforms, computational capacities and functionalities. The network heterogeneity and the ubiquity of IoT devices introduce increased demands on both security and privacy protection. Therefore, the cryptographic mechanisms must be strong enough to meet these increased requirements but, at the same time, they must be efficient enough for the implementation on constrained devices. In this paper, we present a detailed assessment of the performance of the most used cryptographic algorithms on constrained devices that often appear in IoT networks. We evaluate the performance of symmetric primitives, such as block ciphers, hash functions, random number generators, asymmetric primitives, such as digital signature schemes, and privacy-enhancing schemes on various microcontrollers, smart-cards and mobile devices. Furthermore, we provide the analysis of the usability of upcoming schemes, such as the homomorphic encryption schemes, group signatures and attribute-based schemes.

1. Introduction

Nowadays, the Internet of Things (IoT) is a widely-discussed topic among researchers, engineers and technicians. IoT tends to be the next wave of innovation and there are many definitions of the IoT paradigm. For example, IoT can be defined as a highly interconnected network of heterogeneous entities such as tags, sensors, embedded devices, hand-held devices and back-end servers. IoT provides new services and applications that can be deployed in smart homes, transport applications (e.g. Vehicular Ad hoc Networks - VANETs), smart metering, smart grid, etc. Fig. 1 depicts the example of the IoT environment and shows some technologies and appliances that can be used in IoT.

The machine-to-machine and machine-to-human communications are usually based on IP protocol which can cause billions of IoT objects become a part of the Internet. Therefore, the security in IoT has to be addressed due to the high possibility of security risks such as eavesdropping, unauthorized access, data modification, data forgery and unauthorized remote tampering with devices. For example, attackers can turn on smart devices and heating systems to trigger a collapse of the power grid. Furthermore, attacks against routing protocols can be performed in IoT infrastructure and applications, for example, Sybil attacks [1], the sinkhole attack [2].

Security solutions designed for IoT environments have to deal with heterogeneous IoT entities with various hardware specifications. In IoT, the most spread devices are usually resource-constrained devices because of their low cost. These devices usually employ Constrained Application Protocol (CoAP) [3] at the application layer. The security solutions in IoT have to provide the authentication and authorization of IoT nodes (things, users, servers, objects) and data authenticity, confidentiality, integrity and freshness. The security solutions are usually implemented at network, transport and application layers in IoT. Fig. 2 depicts the IoT layers and the security protocols that can be used in IoT, for example, IPSec, Host Identity Protocol(HIP), Transport Layer Security (TLS) protocol, Datagram Transport Layer Security (DTLS) protocol and Slim Extensible Authentication Protocol Over Local Area Networks (SEAPOL). For example, Extensible Authentication Protocol (EAP) messages that ensure Point-to-Point authentication at the link layer can be transfered over SEAPOL or Trust Extension Protocol for Authentication of New deployed Objects and sensors through the Manufacturer (TEPANOM) [4]. Nevertheless, this paper does not aim at the security and authentication protocols at the link and physical layers.

Besides the basic security properties, privacy has to be addressed in IoT as well. Many IoT services and applications provide sensitive and personal information that are exposed, and can be misused by an attacker. Unsecured sensitive data can leak to third

* Corresponding author. Tel.: +420541146963.
E-mail address: malina@feec.vutbr.cz (L. Malina).

http://dx.doi.org/10.1016/j.comnet.2016.03.011
1389-1286/© 2016 Elsevier B.V. All rights reserved.

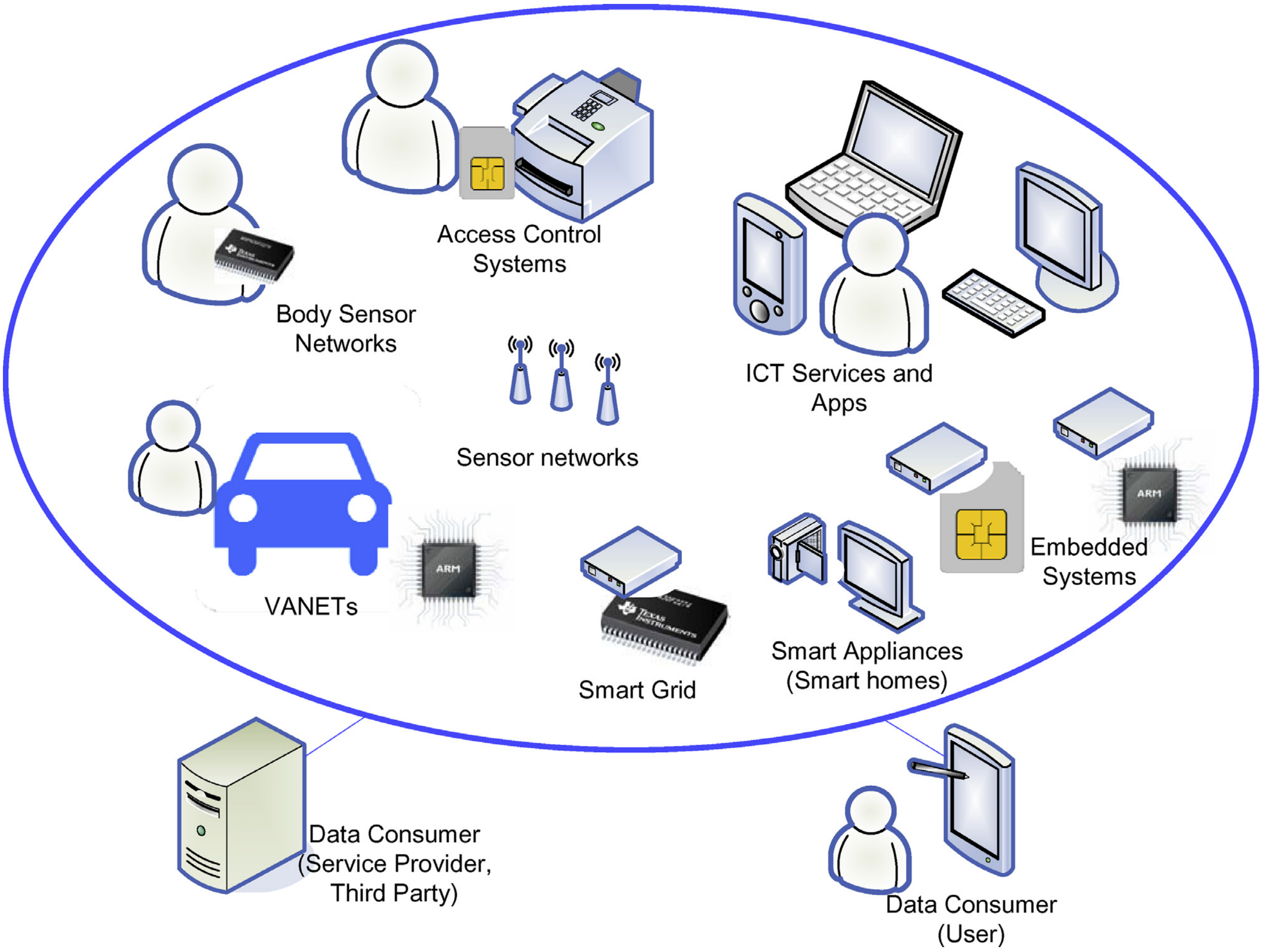
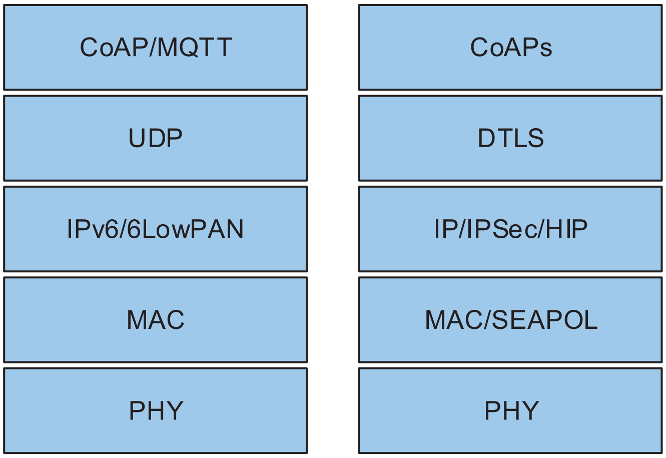


Fig. 1. Technologies and applications in the Internet of Things environment.



CoAP/MQTT	CoAPs
UDP	DTLS
IPv6/6LowPAN	IP/IPSec/HIP
MAC	MAC/SEAPOL
PHY	PHY

Fig. 2. The Internet of Things layers connected with the security protocols.

parties. The concept of privacy may differ but it should protect user's personally identifiable information and keep a certain degree of anonymity, unlinkability and data secrecy.

A lot of privacy-preserving solutions are designed for powerful computers and nodes in the Internet. The privacy-preserving solutions are usually based on computationally expensive cryptographic primitives, such as bilinear pairing, exponentiation of big numbers. Due to this fact, it is still an open challenge to design a secure, efficient and privacy-preserving solution for the IoT that works mostly with the restricted devices. The main goal of this work is to show how common cryptographic primitives are demanded on various devices and show the perspective of some privacy-preserving techniques in IoT.

This article presents the memory limitations and a performance analysis of cryptographic primitives that are measured on the various devices which are used in IoT. Furthermore, we discuss the applicability and limitations of privacy enhancing protocols and schemes. The main purpose of this paper is to construct specified knowledge in privacy-preserving and cryptographic techniques used in the IoT services. The contribution of this work is twofold:

- We present the performance of widely-used cryptographic primitives on various devices and discuss their memory limitations. We focus mostly on operations which are used in security and cryptographic solutions employed in IoT. We implement and measure these operations on various platforms such as microcontrollers, chip cards and ARM devices.
- We discuss and evaluate the privacy-preserving techniques and schemes in IoT. We implement and measure chosen schemes on various platforms (a chip card, an ARM device, PC). We provide interesting insights about which privacy preserving techniques are better to use in the IoT environment. We believe that this work can help with future research based on privacy-preserving mechanisms in the IoT environment.

The rest of this paper is organized as follows:Section 2 describes an overview of IoT security and presents related works.

Section 3 outlines the performance and memory limitations of common cryptographic primitives measured on various devices. Section 4 evaluates current privacy-preserving techniques and discuss their perspective in IoT. Section 5 outlines our conclusions.

2. Security and privacy in IoT

The IoT paradigm has been described in many papers, for example, [5–7]. We define IoT as a highly interconnected network of heterogeneous entities. In our paper, we focus mainly on the evaluation of cryptographic primitives and privacy-preserving solutions that can be used in IoT. In this section, we present the overview of secure communication, authentication and key establishment solutions in IoT and discuss related works dealing with an assessment of cryptographic schemes in IoT.

2.1. Overview of secure communication, authentication and key establishment solutions in IoT

There are many papers which deal with secure communication in IoT and present some new approaches. For example, Weber et al. [8] present a novel identity and access management approach for Future Internet that contains things, people and services. They focus on end-to-end secure communication and user privacy issues. Their solution is based on using a trusted personal device called Minimal Entity. Nevertheless, these devices need to employ Trusted Platform Modules (TPM). Brachmann et al. [9] deal with end-to-end transport security in the IP-based IoT that employs resource-constrained devices and high-performed machines. They propose approaches that ensure end-to-end security between two devices by mapping Transport Layer Security and Datagram Transport Layer Security [10] (TLS/DTLS) in a Low-power and Lossy Network (LLN) with a back-end service that does not support CoAP (Constrained Application Protocol) [3]. Raza et al. [11] present a solution called Lithe which provides an integration of DTLS and CoAP for IoT. They propose a DTLS header compression scheme that enables to reduce the energy consumption by leveraging the 6LoWPAN standard. The DTLS header compression scheme reduces the number of transmitted bytes and maintains the end-to-end security properties. Bonetto et al. [12] propose a secure end-to-end communication scheme between IoT devices and ICT devices. The authors suggest using trusted unconstrained devices for the offloading of computationally intensive operations. These works mainly focus on secure communication but do not deal with authentication with key establishment mechanisms and the privacy protection of users.

Authentication and key establishment solutions for IoT have been investigated in papers such as [13–15]. Hummen et al. [16] deal with the constraints of IoT objects and processing the certificate-based DTLS handshakes. They propose three ideas that are based on a pre-validation, a session resumption and a handshake delegation to reduce computation and memory overheads. Kothmayr et al. [17] deal with a DTLS handshake and X.509 certificates containing RSA keys. The authors implement and measure these mechanisms on sensor nodes featuring Atmel microcontrollers and TPM chips and investigate the time, memory and energy consumptions of a RSA based handshake and an ECC based handshake.

Porambage et al. [13] propose a lightweight authentication and keying mechanism for Wireless Sensor Networks (WSNs) in the distributed IoT applications. Their authentication scheme uses implicit certificates and provides application level end-to-end security. Their solution based on Elliptic Curve Cryptographic (ECC) provides the distribution of credentials, authentication and key establishment in mutual communication between WSN nodes and the users employed in IoT. Nevertheless, the solution is based on Certificate Authority (CA) and authentication takes more than 8 s on TelosB sensor nodes. Vucinic et al. [14] present an object-based security architecture solution that provides confidentiality and authenticity in IoT. The solution called OSCAR is based on Elliptic curve cryptography public key operations (ECDSA signatures) and AES in the Counter with CBC-MAC mode (AES-CCM). They compare their proposed security solution which uses the application layer with the Lithe-DTLS protocol presented in [11]. The OSCAR solution provides end-to-end security and shifts cryptographic burden to clients. This approach enables to deploy constrained servers in IoT with many clients. Further, Keoh et al. [18] analyze DTLS for the IoT applications and current IoT security standardization. They argue that device bootstrapping and key management should be standardized to provide a common management interface and enable to employ large-scale IoT. Saied et al. [15] discuss the limitations of existing end-to-end security standards and key agreement schemes. They propose collaborative approaches for key agreement protocols such as the TLS handshake protocol, Internet Key Exchange (IKE) protocol and collaborative key agreement. Constrained devices may delegate expensive cryptographic operations to less constrained nodes in the IoT environment. These papers usually try to present efficient authentication and key agreement. On the other hand, the privacy-preserving solutions are usually more expensive and provide more security properties. The perspective of privacy preserving schemes and their use in IoT have to be addressed as well.

2.2. Related works

There are few papers that somehow analyze and evaluate some privacy, security and cryptographic methods in IoT (e.g. [19–23]). For example, Cirani et al. [19] provide an overview of the security challenges related to the deployment of smart objects in IoT. They discuss various security protocols at network, transport and application layers such as IPSec, Host Identity Protocol and Datagram Transport Layer Security protocol. Further, they discuss lightweight and common cryptographic algorithms that can be used in IoT such as the AES cipher, the TEA cipher, the PRESENT cipher, the Hight cipher, the RSA scheme, the ECDSA scheme and some hash functions. The authors also discuss key agreement protocols, group key distribution, homomorphic encryption and service authorization. Nevertheless, they do not analyze the performance of cryptographic schemes on various devices.

Roman et al. [20] deal with the features and challenges of security and privacy in the distributed IoT. The paper overviews the existing IoT security challenges and outlines an attacker model for centralized and distributed IoT architectures and describes the main security challenges and promising solutions in IoT. Their paper analyzes security challenges such as identity and authentication, access control, network security, privacy, trust management and fault tolerance.

Sicari et al. [21] present a survey paper which deals with security, privacy and trust in IoT. Further, they focus on secure middlewares, mobile security and several ongoing security projects in IoT. The contribution of the European projects on the IoT security is compared. According to their observations, Europen Commission FP7 project called Butler (www.iot-butler.eu) and project iCore (www.iot-icore.eu) try to deal with the key security issues in IoT. Abomhara and Koien [22] provide an explicit survey of the most important security and privacy aspects of IoT. They argue that privacy aspects should be considered very seriously because the IoT applications often work with sensitive data and personal data. There are also papers that focus mainly on privacy in IoT, such as [24,25]. These papers analyze possible threats and challenges.

Nevertheless, these all surveys discuss some current security and privacy-preserving solutions in IoT but do not test the cryptographic schemes on various devices and do not provide the practical results of privacy preserving methods.

The most related work to our article is the paper of Nguyen et al. [23] which presents the analysis of the key-bootstrapping cryptographic protocols in IoT. Their paper discusses and analyzes key establishment and authentication techniques based on asymmetric schemes and symmetric pre-distributed keys. These key bootstrapping solutions are compared by their security, memory, scalability and performance properties. Nevertheless, they only categorize the performance of these solutions into two levels, good and medium performance level and low performance level. In addition, the privacy protection schemes are discussed only in a nutshell. In our article, we provide a more detailed analysis of privacy-preserving methods including practical results.

In the next sections of our paper, we present the performance assessment of the basic cryptographic schemes and primitives that are considered in security solutions designed for the IoT applications. We implement and measure these primitives on various types of devices that have different specifications to get the performance and memory demands of the cryptographic operations. Based on these results and our experimental implementations, we are able to evaluate some existing privacy-enhancing techniques. We present practical results and discuss their suitability for IoT with resource-constrained devices.

3. Cryptographic primitives and schemes on various devices

In this section, we present the performance of widely used cryptographic primitives, schemes and some important modular arithmetic operations which are used in security solutions implemented in IoT. Besides the performance of the operations, we also discuss their memory requirements.

We implement and measure the operations on various platforms such as microcontrollers, chip cards, smartphones that are used in the IoT environment. We implement the most used cryptographic primitives and schemes such as Advanced Encryption Standard (AES), Secure Hash Algorithms (SHA-1 and SHA-2), asymmetric encryption and signature scheme RSA, random number generator functions and elliptic curve point multiplication used in ECDH and ECDSA schemes. These cryptographic primitives and operations are employed in many IoT security solutions. For example, AES is used for data encryption in the DTLS protocol [10], the Lithe-DTLS protocol [11] and the OSCAR solution [14]. Random number generator and hash functions are used for secret key derivation functions and are employed in many authentication and key establishment protocols. RSA and modular arithmetic operations are used in strong authentication schemes, key establishment protocols (DH, ECDH) and privacy preserving protocols such as [26–28]. The tested cryptographic primitives and schemes are set as follows:

- AES 128b - an encryption of a 128-bit plaintext by the symmetric cipher AES with a 128-bit key in the ECB mode.
- SHA1 4256b - a hash function SHA1 with a 4256-bit plaintext.
- SHA2 8448b - a hash function SHA2 with a 8448-bit plaintext.
- RSA ver/enc 1024b - a RSA exponentiation operation with a 1024-bit modulo, a 1024-bit base (data) and a small public exponent ($e = 65537$). This operation is used for data encryption or for the verification of a RSA signature.
- RSA sig/dec 1024b - a RSA exponentiation operation with a 1024-bit modulo, a 1024-bit base (data) and a private exponent (1024-bits). This operation is used for data decryption or for RSA signing.
- RSA ver/enc 2048b - a RSA exponentiation operation with a 2048-bit modulo, a 2048-bit base (data) and a small public exponent ($e = 65537$). This operation is used for data encryption or for the verification of a RSA signature.
- RSA sig/dec 2048b - a RSA exponentiation operation with a 2048-bit modulo, a 2048-bit base (data) and a private exponent (2048-bits). This operation is used for data decryption or for RSA signing.
- RND 160b - a random number generator function producing a 160-bit random number.
- RND 560b - a random number generator function producing a 560-bit random number.
- ECPM 128b Fp - an elliptic curve point multiplication operation with 128-bit Fp elliptic curves.

We measure the performance of these primitives on various devices, namely, microcontrollers of family MSP430fX, programmable chip cards (java and Multos) and devices with ARM processors to get an overview of how the cryptographic primitives affect the IoT applications and services. For example, MSP430fX microcontrollers are widely used in devices employed in the smart grid systems (smart meters), home automation systems (smart thermostats, smart air condition controllers), industrial embedded systems and sensor networks. The smartcards are used in access control systems or as SAM (Secure Access Module) in embedded devices or in ICT devices which need a secure module. The ARM platform is often used, for example, in industrial embedded systems and vehicular ad hoc network systems. Also many handheld devices (smartphones, tablets etc.) contain ARM controllers. The devices have different technical specifications which are summarized in Table 1.

A delay caused by processing the cryptographic overhead can negatively affect the quality of services in some IoT applications. Therefore, we define a threshold T = 300 ms that is the maximum latency for cryptographic operations and schemes tolerated by real-time IoT applications. Hence, IoT applications can be divided into two groups:

- Real-time IoT applications ($time =< \mathrm{T}$), for example, patient monitoring applications in body sensor networks that must send vital data in real time, some applications in Vehicular Ad hoc Networks (VANET) that send real-time notifications (break alerts, proximity alerts) to drivers on the road.
- Non-real-time IoT applications ($time > \mathrm{T}$), for example, power consumption monitoring in smart grid, traffic jam monitoring in VANET.

3.1. Performance of cryptographic primitives on resource-constrained devices

Resource-constrained devices are considered as the most employed devices in the IoT infrastructure. In this subsection, we present the performance results of cryptographic primitives that are implemented on resource-constrained devices, namely, 8 MHz microcontroller, 20 MHz microcontroller, 30 MHz java card and 33 MHz Multos card. The technical specifications of these devices are in Table 1.

The implementations of cryptographic functions on the microcontrollers are written in the C programming language (C). We wrap some existed libraries such as LibTomCrypt (libtom.net), OpenSSL (openssl.org) and PolarSSL (polarssl.org). We do not use any external crypto-accelerator hardware modules. Our test application for the java card is written in the JAVA language (Java Card Open Platform v2). The chosen java card provides some cryptographic functions but it does not contain any modular arithmetic functions with big integers. On the other hand, the Multos card with the MULTOS card operation system provides both crypto APIs and some modular arithmetic functions. Our test application for the Multos card is written in C.

Table 1
Technical specifications of devices used.

Designation	Device	Processor	RAM size (RAM)	Storage size (ROM / EEPROM / Flash)
8 MHz Microcontroller	ultra-low-power microcontroller MSP430F149	16-bit CPU with 8 MHz	60 kB	60 kB
20 MHz Microcontroller	microcontroller MSP430F6638	16-bit CPU with 20 MHz	18 kB	256 kB
30 MHz java card	smartcard NXP JCOP CJ3A080v24	16-bit CPU with 30 MHz	6 kB	200 + 80 kB
33 MHz Multos card	smartcard ML3-36k-R1	16-bit CPU with 33 MHz	1088 + 960 B	280 + 60 kB
700 MHz ARM	single board computer Raspberry Pi model B+	32-bit ARM11 Single-core with 700 MHz	512 MB	8 GB
2260 MHz ARM	smartphone Nexus 5 LG	32-bit ARMv7 Quad-core with 2260 MHz	2 GB	16 GB

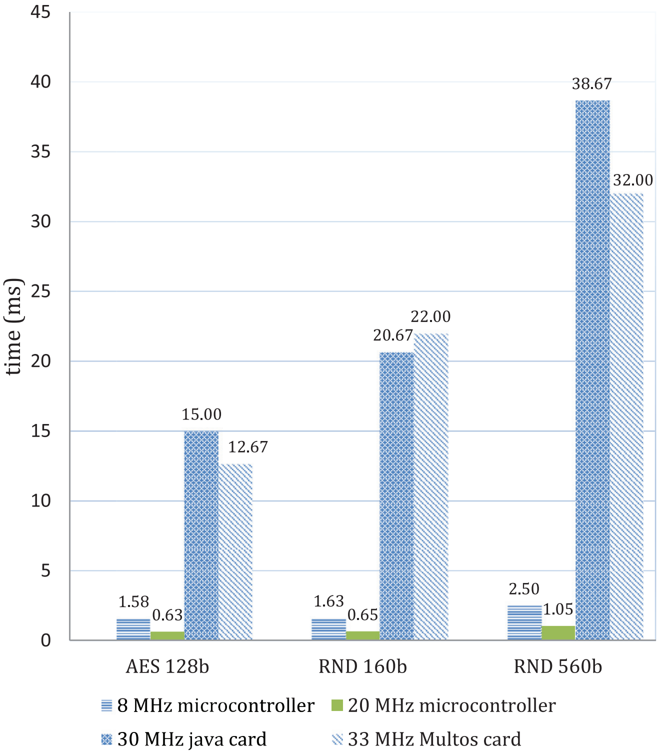


Fig. 3. The execution times of AES 128b, RND 160b and RND 560b operations on resource-constrained devices.

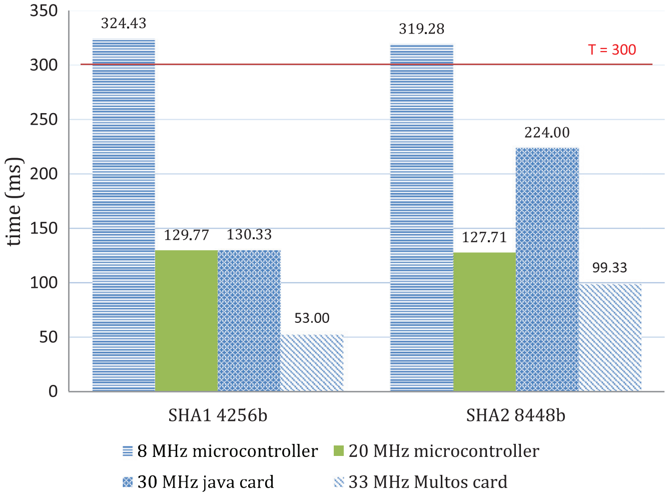


Fig. 4. The execution times of SHA1 4256b and SHA2 8448b operations on resource-constrained devices.

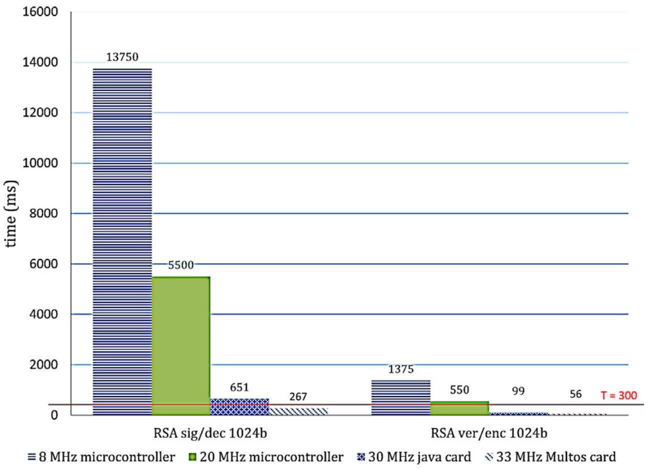


Fig. 5. The execution times of RSAsig/dec and RSAver/enc operations on resource-constrained devices.

The methodology of the measurement is different for microcontrollers and for smartcards. The performance of the cryptographic operations on the microcontrollers is measured as the number of cycles and this number is recomputed on the execution time (1 cycle takes 1μs on a 1 MHz processor). The execution times of the operations on smartcards are average values computed from 10 iterations.

In Fig. 3, the execution times of the AES-128b, RND-160b and RND-560b operations on the resource-constrained devices are depicted. These operation takes only few milliseconds on these devices. The AES encryption operation of one 128-bit data block and the random number generation operations of 160-bit or 560-bit numbers are more efficient on microcontrollers than on smartcards. The initialization of card's operation system and APDU communication between a chip card and a reader device causes time delay. Therefore, chip cards with higher CPUs frequencies than microcontrollers need more execution time for some cryptographic operations in comparison with microcontrollers. Nevertheless, the measurement shows that AES and random number generator functions are efficient and can be implemented into the IoT security solutions that are run on constrained devices.

Fig. 4 depicts the execution times of hash functions SHA1-4256b and SHA2-8448b on microcontrollers and smartcards. The SHA-1 and SHA-2 functions take from tens to hundreds milliseconds on these constrained devices. Many authentication and cryptographic schemes are based on hash functions. The use of such schemes that are performing many hash functions or hashing the large data structures (several kB) can be difficult and problematic in the IoT infrastructure that employs constrained devices.

Fig. 5 shows the execution times of RSAsig/dec and RSAver/enc operations on the low-performance devices. The RSA scheme uses modular exponentiation operations. The RSA operations with public keys take hundreds milliseconds on microcontrollers. The RSA operations with private keys (e.g. a 1024-bit exponent) take several seconds on microcontrollers. The java card and Multos card provide direct RSA APIs that are optimized. Due to this fact, the RSA operations on these smartcards take from tens to hundreds milliseconds.

Modular arithmetic operations such as modular multiplication or modular exponentiation take from several tens to hundreds milliseconds on common smartcards. For example, modular multiplication with 1024-bit numbers takes about 546 ms and modular multiplication with 2048-bit numbers takes about 998 ms on the java card. Nevertheless, some smartcards offer co-processors and direct functions to enhance the performance of some modular arithmetic operations and asymmetric cryptographic operations such as RSA and ECC operations. For example, one ECDH operation with a 128 F_p elliptic curve takes about 104 ms on the java smartcard. Further, the Multos card which supports the big number operations needs only 28 ms to compute one 1024-bit modular multiplication and the Multos card ML2-80K-65 needs only 58 ms to compute one modular exponentiation with a 1024-bit modulo, a 1024-bit base and a 160-bit exponent.

Many symmetric ciphers, for example, AES, are fast enough to be implemented into security solutions that run on constrained devices in the IoT infrastructure. On the other hand, security solutions based on asymmetric cryptographic operations, for example, RSA, ECDH, ECDSA, and big integer modular arithmetic operations (multiplication, exponentiation) need more time to execute, for example, from tens to hundreds milliseconds on constrained devices. For example, RSA signature scheme that takes several seconds on microcontrollers is not suitable for real-time IoT applications (e.g. patient monitoring). On the other hand, smart meters with constrained microcontrollers that usually send power consumption data can use RSA signing because the data are sent only few times per day.

3.2. Performance of cryptographic primitives on middle and high-performed devices

This subsection presents the performance results of cryptographic primitives and schemes that run on devices with ARM chips having several hundreds MHz which are used in embedded devices, mobiles or control devices. We assume that devices based on the ARM platform are also widely used in the IoT infrastructure. These devices are much more computational powerful and enable to use more cryptographic libraries and functions than microcontrollers or smartcards. In our measurement, we use the devices with 700 MHz and 2260 MHz ARM chips. Their specifications are described in Table 1. The 700 MHz ARM device is a single-board computer with a Linux OS (Raspbian) where we run our test application written in the JAVA programmable language. To obtain the fast modular arithmetic operations, we call a C math library, namely GNU Multiple Precision Arithmetic Library (GMP) via Java Native Interface (JNI). The 2260 MHz ARM device is a smartphone with Android OS (Android 4.4 KitKat). Our test application is written in the JAVA programming language by using the Android Software Development Kit (SDK). The platforms of both ARM devices provide many crypto APIs and modular arithmetic operations via big integer APIs.

Fig. 6 depicts the execution times of AES-128b encryption, hash functions SHA1-4256b and SHA2-8448b on the ARM devices. These operations take few milliseconds on the single-board computer with 700 MHz ARM. On the smartphone with 2260 MHz ARM (Quad-core), these operations take about one hundred μs.

Fig. 7 shows the execution times of RND, RSAsig/dec, RSAver/enc and ECDH operations on the ARM devices. The smartphone is able to compute the RSA operations within hundreds μs. The single-board computer needs several tens milliseconds for computing the RSA operations. The random number generator functions generate secret random numbers within few milliseconds. The execution time depends on a generation method which is used in a device.

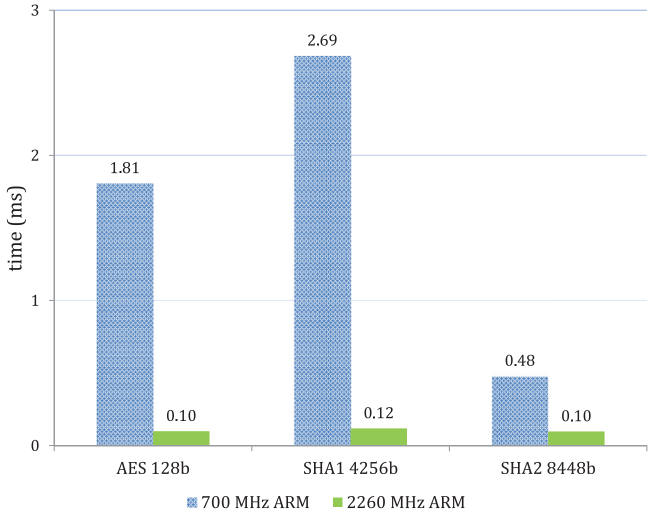


Fig. 6. The execution times of AES 128b, SHA1 4256b and SHA2 8448b operations on middle and high-performed devices.

On the ARM devices, modular arithmetic operations such as modular multiplication or modular exponentiation take from hundreds μs up to tens milliseconds. The times depend on the sizes of inputs, a programing language and a cryptographic/math library. For example, modular multiplication with 2048-bit numbers takes about 0.1 ms and modular exponentiation with a 2048-bit modulus and a 160-bit exponent takes 17 ms on the 700 MHz ARM employing the GMP library written in C.

Our measurement proves that robust security solutions which include asymmetric cryptography and big integer modular arithmetic operations can be implemented on many ARM devices deployed as the IoT nodes. These operations takes several ms and are easy to implement. Nevertheless, some cryptographic operations such as bilinear pairing operations that are widely used in many security privacy-preserving, identity based or group signature schemes are too much computationally expensive. The bilinear pairing operations take from several hundreds milliseconds to few seconds on ARM devices. For example, one asymmetric bilinear pairing operation (175-bit curves) takes about 2.4 s on the smartphone (2260 MHz ARM). Thus, pairing based cryptographic schemes are not suitable for the IoT infrastructure with the various devices that are restricted or are based on less performed ARM platforms. The optimization techniques such as the batch verification and the pairing precomputation can reduce the total number of pairing operations but the IoT security solutions should avoid using the pairing based schemes. Only computationally powerful IoT nodes are able to compute expensive pairing operations in reasonable time.

3.3. Memory requirements of cryptographic primitives

Besides long execution times of some cryptographic operations, the security solutions have to deal with memory constrains of devices in IoT. RAM memory and code size requirements of the cryptographic schemes and primitives are various. Obviously, ARM devices and many smart cards provide enough RAM and a storage memory for cryptographic primitives. These platforms usually have larger RAM and a storage memory than most of microcontrollers. Nevertheless, the microcontrollers with small RAM and a storage memory are usually cheaper and therefore, these devices are widespread in the systems with a large number of nodes, for example, home automation controllers, sensors, smart meters, etc.

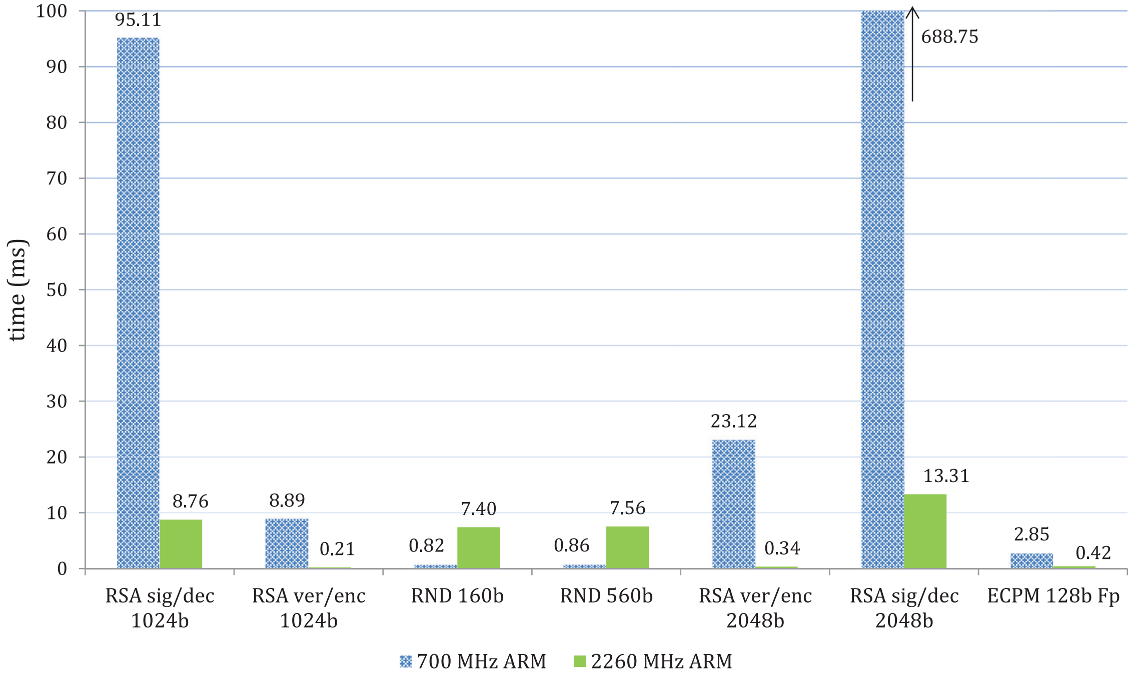


Fig. 7. The execution times of RND 160b, RND 560b, RSA sig/dec 1024b, RSA ver/enc 1024b, RSA sig/dec 2048b, RSA ver/enc 2048b and ECPM 128b Fp operations on middle and high-performed devices.

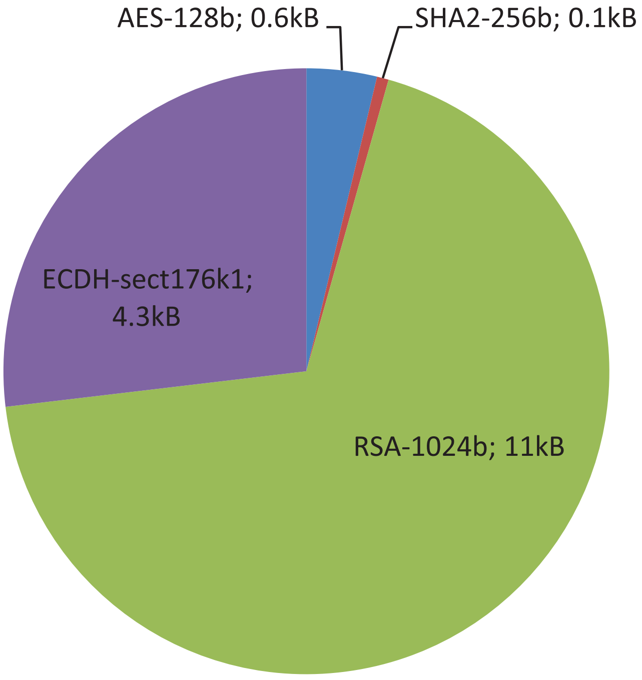


Fig. 8. The RAM consumption of cryptographic primitives on MSP microcontrollers.

However, a small RAM memory is usually problematic for some security schemes that are implemented on those devices. Fig. 8 depicts the RAM consumption of our AES, RSA, ECDH and SHA2 implementations on MSP microcontrollers.

In the following text, we describe the RAM and storage memory requirements of our test implementations written in C on the MSP microcontrollers:

- Asymmetric schemes - our RSA implementation takes almost 249 kB in a storage memory due to many functions wrapped from libraries (LibTomCrypt, GMP) and uses approximately 11 kB of a RAM memory due to large byte array structures for input parameters (1024–2048 bits per one) and variables. Therefore, the complete implementation of the RSA scheme is not feasible for microcontrollers with a small RAM memory (< 10 kB). Further, we implement big integer modular arithmetic and elliptic curve (EC) operations by wrapping the OpenSSL library. The total code size of our implementation is 12412 bytes. RAM memory usage depends on the size of elliptic curves and the types of the arithmetic methods. Nevertheless, the solutions that use ECC and modular arithmetic operations need microcontrollers with a RAM memory at least 4 kB.
- Symmetric ciphers - the ciphers usually take from hundreds bytes to few kB in a storage memory and use approximately from tens to hundreds B of a RAM memory. For example, AES wrapped from the LibTomCrypt library takes 550 B in RAM. There are AES implementations that are optimized for RAM memory usage but the total number of cycles for one 128-bit encryption can be higher. Nevertheless, many ciphers such AES, XTEA, Noekeon can be implemented into microcontrollers with a small RAM memory, for example, 1 kB RAM, because the operations of symmetric ciphers are repeated in rounds and work with smaller keys (e.g. 128-bits) than asymmetric ciphers (e.g. 1024-bits).
- Hash functions - these functions usually take few kB in a storage memory and use approximately tens to hundred B of a RAM memory, for example, 107 B in RAM by using the SHA-256 function wrapped from the LibTomCrypt library. A small RAM consumption of the SHA2 function can be explained by using small variables (32-bits) and simple operations that are repeated in rounds. Many hash functions can be implemented into microcontrollers with a small RAM memory, for example, 1 kB RAM.

4. Perspective of privacy-preserving techniques in internet of things

A lot of the IoT services can sense and collect sensitive data such as an actual user location, personal data, vital and medical data and so on. The absence of privacy protection functions and approaches may cause serious privacy leakage. In the following subsections, we analyze and evaluate the privacy-preserving techniques and their use in the IoT infrastructure that may employ restricted devices.

Our analysis is based on our evaluation of cryptographic primitives on various devices which is presented in the previous section. We focus especially on cryptographic solutions and schemes that provide privacy. These cryptographic solutions are usually based on standard cryptographic operations (data encryption, hash functions) and modular arithmetic (multiplication, exponentiation). In addition, we implement selected privacy-preserving solutions and measure their time execution on various devices.

The following subsections present these types of privacy-preserving techniques:

- General approaches for data privacy and *k*-anonymity - this group of techniques represents basic and naive approaches that use, for example, data masking, encryption and pseudonyms to enhance privacy protection.
- Homomorphic encryption - this technique represents the advanced encryption schemes that enable to work with ciphertexts without the need of their decryption. The privacy protection is enhanced because user's data remain in secret at the side of third parties and service providers.
- Group signatures and ring signatures - these two techniques represent advanced digital signature schemes that provide security and signer anonymity.
- Attribute based signatures (ABS) and attribute based encryption (ABE) - ABS represents advanced digital signature schemes that enable to prove the possession of an attribute (e.g. user's age, user's membership) without revealing user's identity. ABE enables users to encrypt and decrypt data based on their attributes.

4.1. General approaches for data privacy and k-anonymity

The basic level of privacy can be achieved by using the encryption of sensitive data. The encryption of transmitted data prevents the passive attackers who are eavesdropping on the communication from getting the content. Nevertheless, this approach can be applied in systems where two communication nodes trust each other and share a common secret key for the data symmetric encryption (AES, XTEA, IDEA, ...). This secret key must be securely established or pre-distributed. Asymmetric encryption schemes such as RSA, ElGamal enable to encrypt the data by public keys and simplify the key distribution in systems. Nevertheless, the computational expensiveness of these schemes prevents their spread in the IoT systems with constrained devices. The asymmetric encryption schemes are usually used for the encryption of small messages and secret keys. The ElGamal scheme is probabilistic which means that a single plaintext can be encrypted to many possible ciphertexts. This property provides unlinkability which is often needed in the privacy preserving solutions.

Another privacy protection approach is *k*-anonymity [29] which guarantees that each released record relates to at least *k* records. The *k*-anonymity approach protects against identity disclosure. Nevertheless, this approach does not sufficiently provide protection against attribute disclosure as shown in [30]. Huang et al. [26] deal with the combination of context aware access control and data transformation to protect privacy. Their solution uses a context aware *k*-anonymity that manipulates the identifiers of the records until each record cannot be distinguished from *k*-1 records. A data publisher transforms the raw data by using his/her privacy settings, for example, his/her sensitive data can be masked or substituted with ambiguous values. To protect the content of data, public key cryptography is used for the encryption of the transmitted information. Further, the users can control which of their personal data are processed and collected and who can take their data. Ukil et al. [27] propose a negotiation-based privacy-preserving scheme for the centralized IoT architecture. Privacy and utility are negotiated between a data consumer and a data producer. They describe a dynamic masking technique for enforcing privacy. Their solution employs traditional IT security primitives such as SSL/TLS. Further, Shen et al. [31] propose secure and privacy-preserving location sharing mechanism. It employs the bloom filter [32] that is used for hiding sensitive private information. Celdran et al. [33] propose a context-aware policy framework for preserving user privacy in IoT. The framework is based on semantic rules that form policies such as operational, authorization, cloaking location, hiding location, granularity location and closeness location. This approach differs from the previous ones due to that the framework employs a trusted third party that manages users' privacy and their location data. However, the proposed framework does not contain anonymity and hashing policies to disguise the identities of users. Recently, Gope and Hwang [34] propose an anonymous authentication scheme which is designed for a distributed IoT system architecture. Their scheme can ensure properties like sensor anonymity, untraceability, resistance to replay attacks and cloning attacks, and mutual authentication. The scheme is lightweight due to using the hash functions. Nevertheless, the scheme does not solve data authenticity and privacy or some other security and privacy properties (non-repudiation, revocation, etc.).

Data privacy and *k*-anonymity can be enough in some IoT applications and services which require specific security and privacy properties. Nevertheless, it can be hard to achieve all security and privacy properties such as anonymity, confidentiality, message authentication, non-repudiation, unlinkability, tracebility or user revocation by one approach. The security and privacy-preserving solutions then must combine more cryptographic and privacy-enhancing techniques, such as homomorphic encryption, group signatures or attribute-based signatures that are described in the following subsections.

4.2. Homomorphic encryption

Homomorphic encryption allows the users to encrypt sensitive data and enables to process these encrypted data without their decryption. These encrypted data can be processed by another party without revealing what information is inside. There are two basic types of homomorphic encryption schemes: Partially Homomorphic Encryption(PHE) and Full Homomorphic Encryption (FHE). There are several partially homomorphic encryption systems such as Paillier [35] or Benaloh [36]. Nevertheless, some works such as [37–39] show that fully homomorphic encryption (FHE) schemes are very computationally and memory demanded. According to the paper [40], homomorphic encryption can be also a part of a secure multi-party computation that creates the new opportunities in the area of development privacy-preserving ubiquitous applications. Further, Sun et al. [41] propose a multiplication homomorphism method that is used as a privacy protection solution in IoT services.

Generally, homomorphic encryption can provide data privacy during data aggregation services (smart grid services [42], WSN services [43], healthcare monitoring with an IoT platform [44], IoT data collection services [45]). These solutions are usually based on the Pailler's homomorphic encryption scheme [35]. This PHE scheme provides the additive property. The product of two ciphertexts is equal to the sum of two corresponding plaintexts after the decryption of the product. This encryption enables to sum encrypted data without a private key. In addition, the Pailler's homomorphic encryption scheme enables the addition and multiplication of a plaintext by a constant value. These properties are useful in privacy-preserving data aggregation services. The Pailler's scheme with several modular arithmetic operations in encryption (two exponentiation and one multiplication) and in decryption (one exponentiation, two multiplication, one division) is more expensive than the RSA scheme (one exponentiation). The equations for encryption and decryption are defined as follows:

$$\begin{aligned} c &= g^m \cdot r^n \mod n^2, \\ m &= (c^\lambda \mod n^2 - 1)/n \cdot \mu \mod n, \end{aligned} \tag{1}$$

where c is a ciphertext, m is a message in a plaintext, r is a random number $r \in Z_n^*$, (n, g) is a public key and (λ, μ) is a private key.

Homomorphic encryption schemes can be the useful tools for the applications using the high-performed devices or cloud storage solutions but currently, the application of this technology in IoT with resource-constrained nodes is not practical due to expensive operations and large sizes of keys, parameters or ciphertexts.

4.3. Group signatures and ring signatures

Common digital signature schemes are usually linkable and traceable to a user identity. If a user identity is decoupled from a verification procedure then the privacy, authentication and unlinkability of a user can be ensured. Group Signature (GS) schemes allow the users to authenticate themselves on behalf of a group without using certificates or user identities. A user who is a member of a group can sign a message behalf of the group and sends it anonymously to a verifier. The signature is produced by using a group secret member key and is verified by one public group key that is publicly spread in the system.

Group signature schemes could be used in many privacy-preserving services and applications. GS firstly introduced in 1991 by Chaum [46] have been investigated by many researchers who presented many schemes, for example, the scheme proposed by Boneh, Boyen and Shacham [47], by Delerablée and Pointcheval [48], the scheme proposed by Boyen and Waters [49] or Libert, Peters and Yung's scheme [50]. Many papers, for example, [51–54], try to apply group signature schemes in Mobile Ad-hoc Networks (MANETs), Vehicular Ad hoc Networks (VANETs) and other broadcast communication systems where privacy and anonymity of senders are needed. These vehicular networks and ad hoc systems can be a subset of the IoT infrastructure.

Nevertheless, group signature schemes are not suitable for constrained devices due to many expensive operations such as modular exponentiation and bilinear pairing operations. The signature and verification phases of some group signature schemes take too much time even by using the computationally powerful nodes. For example, the signing phase of the Boneh, Boyen and Shacham scheme [47] takes several seconds on smartphones. Some GS schemes produce larger signatures (e.g. around 6 kB in the scheme [50]) and use longer keys than classic signature schemes such as RSA or ECDSA. Therefore, the bandwidth restrictions of the IoT infrastructure and the memory restrictions of the IoT devices prevent the implementation of group signature schemes in the privacy-preserving IoT services.

Ring Signcryption/Ring Signature (RS) schemes can protect the sender privacy because a receiver only knows that a ciphertext/signature comes from a member of a ring. Li et al. [55] propose a ring signcryption scheme for a heterogeneous IOT data transmission between sensors and a server. Their scheme achieves confidentiality, integrity, authentication, non-repudiation and anonymity without the need of certificates. The signcryption takes n+2 point multiplications and few additions, hash functions and XOR operations. For example, $n = 100$ members in the ring need about 80 s to perform the signcryption on the MICA2 device with the ATmega 128 8-bit processor [55]. The unsingcryption takes n point multiplications, 2 pairing operations and few less expensive operations (hashes, additions, etc.). Therefore, the receiver needs a powerful device (e.g. a server).

4.4. Attribute based signatures and attribute based encryption

Attribute based signature (ABS) schemes allow users to generate signatures with attributes which are satisfying a policy without leaking more information. The users who request some data or services have to generate signatures by using the attributes. These signers remain anonymous and are indistinguishable among all users. The signers are not able to forge signatures with attributes that they do not own. Only a user with the valid attributes is able to endorse the message which is then sent in the IoT infrastructure. More details about the attribute based signature and its application can be found in [56]. The attribute based signatures are often based on the attribute based credentials schemes, such as Idemix [57], U-Prove [58] and the HM scheme [59] that are used in authentication systems where users are proving the possession of the attributes.

Su et al. [28] propose an attribute based signature (ABS) scheme that employs an attribute tree and expresses any policy consisting of AND, OR and threshold gates. The signature generation algorithm needs 2(l+1) exponentiation operations where l is the size of the set of attributes associated with leaves in the attribute tree. The verification of the signature takes 2l bilinear pairing operations and few exponentiation operations that depends on the size of the tree. The length of the signature is 2l+2. Due to the expensive operations (bilinear pairing and exponentiation) in the proposed scheme, it can be applied only in the IoT privacy protected services that employ powerful nodes.

Alcaide et al. [60] propose a decentralized anonymous authentication protocol for users who send data to a data collector in IoT. They combine recent anonymous credentials using Zero-Knowledge Proofs of Knowledge (ZKPK) techniques, secret sharing and threshold cryptography. The protocol contains many exponentiation operations and is appropriate for powerful platforms (standard PCs). Nevertheless, Lin et al. [61] argue that their protocol is insecure. An adversary who impersonates a legitimate user can cheat the data collectors.

The paper [62] provides the performance estimation of selected privacy-preserving schemes, such as proofs of knowledge schemes, Idemix [57], U-Prove [58] and the HM scheme [59]. These schemes take several seconds on smartcards and hundreds milliseconds on smartphones. Therefore, the attribute-based signature schemes can be implemented in privacy-preserving IoT services with nodes that have enough space of memory (RAM and storage). On the other hand, the disadvantages of some attribute-based protocols are a large size of the signature and the need of trusted platform modules. Although, non-pairing based privacy-preserving protocols are usually more computationally efficient than pairing based solutions, proving user's attributes takes more time than computing the common cryptographic primitives such as digital signatures (RSA, ECDSA). The attribute based signatures can be used in some delay-tolerant IoT services that require strong privacy.

Attribute-Based Encryption (ABE) has enhanced the Identity-Based Encryption (IBE) that defines public keys as arbitrary strings, for example, the email address, names etc. ABE does not use an identity as a public key but defines a set of attributes (e.g. roles) that are needed for encryption or decryption. ABE schemes can be based on keys, i.e., Key-Policy ABE (KP-ABE) where the message can be decrypted only by a user that holds the set of the attributes. Ciphertext-Policy ABE (CP-ABE) schemes use policies that are defined over the set of attributes with using conjunctions, disjunctions and threshold gates. ABE schemes are usually computationally expensive due to many pairings but this approach can be

Table 2
Technical specifications of devices used in the comparison of privacy-preserving techniques.

Type	Device	Processor	RAM size (RAM)	Storage size (ROM / EEPROM / Flash)
Chip card	smartcard ML3-36k-R1	16-bit CPU with 33 MHz	1088 + 960 B	280 + 60 kB
Mobile	smartphone Nexus i9250	32-bit ARM 2x 1200 MHz	1024 MB	8 GB
PC	personal computer Lenovo Think station E20	64-bit Intel Xeon CPU 8x 2.53 MHz	8 GB	16 GB

Table 3
Comparison of privacy-preserving techniques.

Type of technique/Scheme	Performance overhead on Chip card	Performance overhead on Mobile	Performance overhead on PC	Memory and communication overhead
HE/Paillier's scheme [35] (1024 b parameters)	Enc = 488 ms; Dec = 746 ms	Enc = 48 ms; Dec = 90 ms	Enc = 20 ms; Dec = 45 ms	ciphertext size = $2l_z$ (e.g. 2048 b)
HE/Paillier's scheme [35] (2048 b parameters)	Enc = 799 ms; Dec = 1213 ms	Enc = 368 ms; Dec = 686 ms	Enc = 146 ms; Dec= 303 ms	ciphertext size = $2l_z$ (e.g. 4096 b)
GS/BBS scheme [47]	-	Sig = 11226 ms; Ver = 33153 ms	Sig = 215 ms; Ver = 518 ms	$3l_{G_1}$ + $6l_p$ (e.g. 1533 b)
GS/DP scheme [48]	-	Sig = 15778 ms; Ver = 32016 ms	Sig = 208 ms; Ver = 516 ms	$4l_{G_1}$ + $5l_p$ (e.g 1444 b)
RS/Li scheme [55] (100 identities)	-	Signcrypt = 13362 ms; Unsigncrypt = 20294 ms	Signcrypt = 510 ms; Unsigncrypt = 580 ms	ciphertext size = $\|m\| + (n+2)l_{G_1}$ (hundreds kB)
ABS/HM scheme [59] (1 attribute)	Sig = 2509 ms; Ver = 2515 ms	Sig = 45 ms; Ver = 44 ms	Sig = 16 ms; Ver = 17 ms	$3l_n$ + $4l_z$ + $1l_c$ (e.g. 4265 b)
ABE/GPSW scheme [64] (AT=10)	-	Enc = 1572 ms; Dec = 38590 ms	Enc = 60 ms; Dec = 500 ms	ciphertext size = $(AT+1)l_{G_1}$ + l_{G_T} (e.g. 2907 b)

useful in some cloud storage applications [63]. Several proposed schemes exist, for example, Key-Policy ABE (GPSW) scheme [64], Water's CP-ABE scheme [65].

4.5. Comparison of privacy-preserving techniques

In this subsection, we provide the comparison of advanced privacy-preserving techniques such as Homomorphic Encryption (HE), Group Signatures (GS), Ring Signatures (RS) and Attribute-Based Signatures (ABS). We choose and compare these cryptographic schemes: the Paillier's homomorphic encryption scheme [35], the BBS group signature scheme [47], the DP group signature scheme [48], the ring signcryption (Li et al.) scheme [55], the attribute based signature (HM) scheme [59] and the attribute based encryption (GPSW) scheme [64].

In the comparison, we assume three types of IoT devices: a resource-constrained IoT device (Chip card with the OS Multos), a medium-performed IoT device (Mobile with Android 4.2) and a high-performed IoT device (PC with Windows 7). All these devices provide the sufficient space of RAM and storage memory for privacy-preserving schemes. Furthermore, the devices offer programmable platforms for loading own applications and libraries. Chosen privacy-preserving schemes are implemented and loaded on these devices. The technical specifications of the devices are listed in Table 2.

The selected schemes are implemented in three programing languages that depend on the device. The Android platform is used for the implementation and the tests on the mobile device. JAVA is used for the implementation and the tests on the PC device and the programming language C is used for the implementation and the tests on the chip card device.

Table 3 presents our experimental results for selected privacy-preserving techniques. We measure the performance overhead on the devices used and we estimate memory/communication overhead. We focus on the time of main operations/phases such as the encryption time (Enc), the decryption time (Dec), the signing time (Sig) and the verification time (Ver), the signcryption time (Signcrypt) and the unsigncryption time (Unsigncrypt). All time values are computed as the mean values from 10 iterations. Due to the complexity of the library that provides the bilinear pairing operations, we do not implement pairing-based solutions on the Chip card device. The techniques provides at least the 80-bit security level. The length l_{G_1} describes the length of the group element $\in \mathbb{G}_1$ (e.g. 171-bits). l_{G_T} describes the length of the group element $\in \mathbb{G}_T$ (e.g. 1026-bits). l_p denotes the length of scalars in modulo p (e.g. 170-bits). l_n denotes the length of the RSA modulo n (e.g. 1024-bits). l_z denotes the length of the scalars of various lengths less than n (e.g. $<$ 1024-bits). l_c denotes the length of the hash used (e.g. 160-bits). l_{ec} denotes the length of the elliptic curve element (e.g. 169-bits). AT denotes the size of an attribute set.

Fig. 9 depicts the performance overhead of selected privacy-preserving techniques on the chip card, the mobile device and PC.

The practical results of the performance overhead show that many of advanced schemes are not suitable for IoT real-time applications with constrained IoT nodes. Especially, the execution times of schemes with many expensive operations such as pairing operations (e.g. 3.5 s on the mobile device used), point multiplication or exponentiation (e.g. 131 ms on the mobile device used) take hundreds milliseconds or seconds on IoT devices such as smartphones or single-board computer units. Therefore, the privacy protection solutions need more computationally powerful nodes that can perform the expensive cryptographic operations. Furthermore, several privacy-preserving schemes need special cryptographic libraries in order to compute operations such as pairings and so on. Thus, the code size and memory demands are higher than with basic cryptographic methods (AES, RSA, SHA-1,...).

Current trends in IoT such as employing wearable devices and low-cost sensors cause that the IoT environment will consist of more restricted devices than powerful devices (laptops, smartphones). Hence, the privacy-preserving techniques must be efficient and easy-to-deploy at the side where the restricted devices are used. For example, the signing data by some anonymous digital signature schemes (group or attribute signature schemes) should be as efficient as possible in some data collection services. The signing phase should not contain bilinear pairings and only the minimal number of expensive operations.

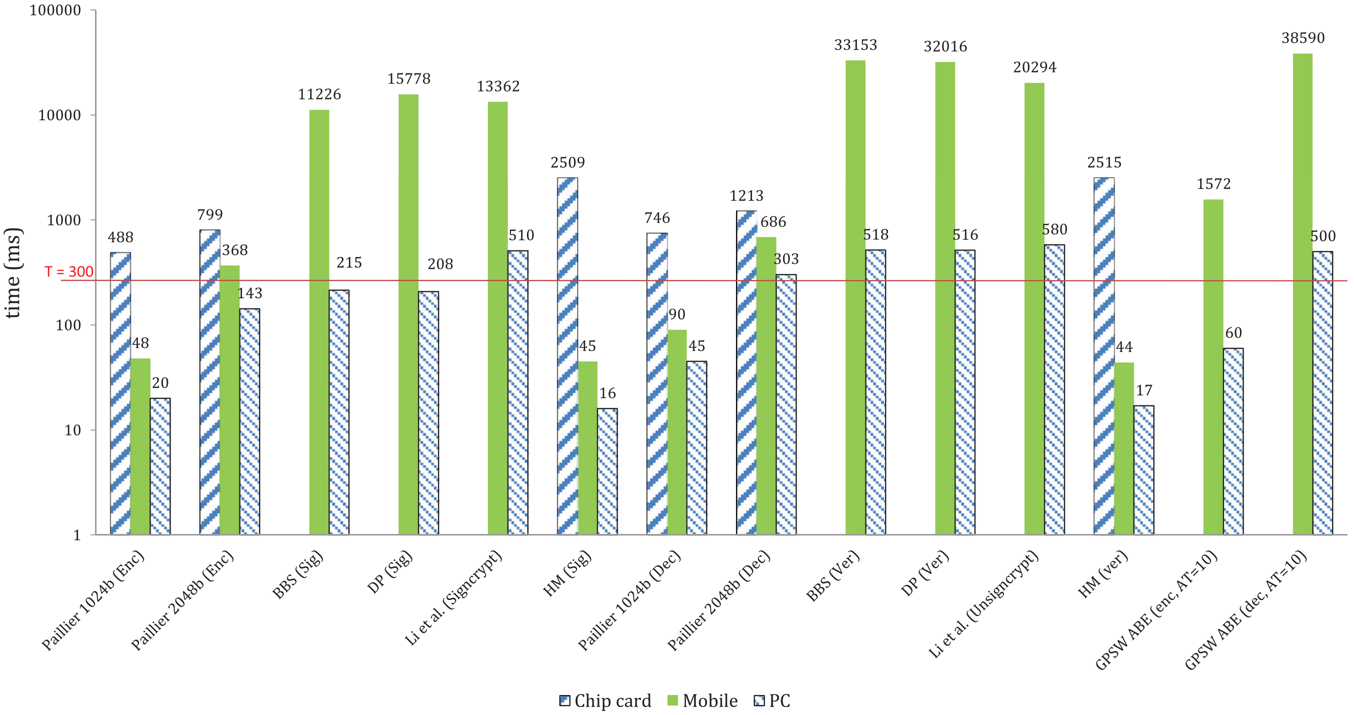


Fig. 9. The execution times of privacy-preserving techniques on the devices.

5. Conclusion

Many works and surveys discuss the security and privacy in the IoT. Nevertheless, only few concrete works present useful and applicable security solutions for IoT. This work presents the performance and memory limitations of current cryptographic primitives and schemes on various types of devices that can be used in IoT. Nowadays, symmetric ciphers and hash functions can be easily implemented into the IoT services that use constrained devices. These functions take only few milliseconds and can run on memory restricted microcontrollers with RAM less than 1 kB. Asymmetric cryptographic schemes and modular arithmetic operations can be used in the IoT services and applications as well. However, the devices should provide at least middle-sized RAM (e.g. $>$ 4 kB) and storage memories (e.g. $>$ 10 kB). Further, the time execution of some asymmetric cryptography functions and operations, for example, RSA signing by a 2048-bit private key, can cause a latency more than hundreds milliseconds on computationally constrained devices such as MSP microcontrollers, smart cards, etc. For example, applications that must sign and send data in real time cannot employ such computational expensive operations on restricted devices.

We also analyze privacy-preserving techniques such as data privacy and *k*-anonymity, homomorphic encryption, group signatures, ring signatures, attribute based signatures, attribute based encryption and their perspective in IoT. Our analysis shows that many strong privacy-preserving solutions are based on proof of knowledge schemes, bilinear pairing operations and employ public key cryptography schemes. IoT with constrained devices with a small RAM memory may have difficulties to employ these privacy solutions. The special cryptographic operations such as bilinear pairings, modular exponentiation and point multiplication have usually high memory and RAM demands due to the need of the external cryptographic libraries. The most computationally expensive cryptographic operation on ARM devices is a pairing operation. For example, one pairing operation with 175-bit curves takes about 2.3 s on a 2.26 GHz ARM device.

The privacy-preserving techniques such as homomorphic encryption, group signature schemes, attribute based signature schemes, attribute based encryption or signcryption schemes need from tens milliseconds to several seconds for their phases, i.e., sign, encrypt, verify, decrypt, on devices such as chip cards and ARM devices. The secure and privacy-preserving IoT applications need a solution that is not based on expensive bilinear pairing operations, produces short signatures and is easy to deploy in memory-restricted devices. The non-pairing attribute based signature schemes seem as a perspective privacy-preserving technique in some IoT applications. On one hand, these schemes can be implemented also on chip cards, SAM modules and other constrained devices. On the other hand, these schemes often need a tamper-proof module and can produce the large sizes of signatures.

Acknowledgement

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

References

[1] K. Zhang, X. Liang, R. Lu, X. Shen, Sybil attacks and their defenses in the internet of things, Internet Things J. IEEE 1 (5) (2014) 372–383.
[2] C. Cervantes, D. Poplade, M. Nogueira, A. Santos, Detection of sinkhole attacks for supporting secure routing on 6lowpan for internet of things, in: Integrated Network Management (IM), 2015 IFIP/IEEE International Symposium on, IEEE, 2015, pp. 606–611.
[3] Z. Shelby, K. Hartke, C. Bormann, B. Frank, Constrained application protocol (coap), draft-ietf-core-coap-18 (work in progress), sl: Ietf 2013.
[4] M. Pawlowski, A. Jara, M. Ogorzalek, Eap for iot: More efficient transport of authentication data – tepanom case study, in: Advanced Information Networking and Applications Workshops (WAINA), 2015 IEEE 29th International Conference on, 2015, pp. 694–699, doi:10.1109/WAINA.2015.53.
[5] L. Atzori, A. Iera, G. Morabito, The internet of things: A survey, Comput. netw. 54 (15) (2010) 2787–2805.
[6] J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of things (iot): A vision, architectural elements, and future directions, Future Gener. Comput. Syst. 29 (7) (2013) 1645–1660.

[7] P. Barnaghi, W. Wang, C. Henson, K. Taylor, Semantics for the internet of things: early progress and back to the future, Int. J. Semantic Web Inform. Syst. 8 (1) (2012) 1–21.

[8] S.G. Weber, L.A. Martucci, S. Ries, M. Mühlhäuser, Towards trustworthy identity and access management for the future internet, in: The 4th International Workshop on Trustworthy Internet of People, Things & Services (Trustworthy IoPTS 2010) co-located with the Internet of Things 2010 Conference, November, 2010.

[9] M. Brachmann, S.L. Keoh, O.G. Morchon, S.S. Kumar, End-to-end transport security in the ip-based internet of things, in: Computer Communications and Networks (ICCCN), 2012 21st International Conference on, IEEE, 2012, pp. 1–5.

[10] E. Rescorla, N. Modadugu, Datagram transport layer security version 1.2(2012).

[11] S. Raza, Lightweight security solutions for the internet of things, Mälardalen University, Västerås, Sweden, 2013 Ph.D. thesis.

[12] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, M. Rossi, Secure communication for smart iot objects: Protocol stacks, use cases and practical examples, in: World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2012 IEEE International Symposium on a, IEEE, 2012, pp. 1–7.

[13] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, M. Ylianttila, Pauthkey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed iot applications, Int. J. Distrib. Sensor Netw. 2014 (2014).

[14] M. Vučinić, B. Tourancheau, F. Rousseau, A. Duda, L. Damon, R. Guizzetti, Oscar: Object security architecture for the internet of things, Ad Hoc Netw. (2014).

[15] Y.B. Saied, A. Olivereau, D. Zeghlache, M. Laurent, Lightweight collaborative key establishment scheme for the internet of things, Comput. Netw. 64 (2014) 273–295.

[16] R. Hummen, J.H. Ziegeldorf, H. Shafagh, S. Raza, K. Wehrle, 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, in: HotWiSec '13, ACM, New York, NY, USA, 2013, pp. 37–42.

[17] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, G. Carle, Dtls based security and two-way authentication for the internet of things, Ad Hoc Netw. 11 (8) (2013) 2710–2723.

[18] S. Keoh, S. Kumar, H. Tschofenig, Securing the internet of things: A standardization perspective (2014).

[19] S. Cirani, G. Ferrari, L. Veltri, Enforcing security mechanisms in the ip-based internet of things: An algorithmic overview, Algorithms 6 (2) (2013) 197–226.

[20] R. Roman, J. Zhou, J. Lopez, On the features and challenges of security and privacy in distributed internet of things, Comput. Netw. 57 (10) (2013) 2266–2279.

[21] S. Sicari, A. Rizzardi, L. Grieco, A. Coen-Porisini, Security, privacy and trust in internet of things: The road ahead, Comput. Netw. 76 (2015) 146–164.

[22] M. Abomhara, G. Koien, Security and privacy in the internet of things: Current status and open issues, in: Privacy and Security in Mobile Systems (PRISMS), 2014 International Conference on, 2014, pp. 1–8, doi:10.1109/PRISMS.2014.6970594.

[23] K.T. Nguyen, M. Laurent, N. Oualha, Survey on secure communication protocols for the internet of things, Ad Hoc Netw. (2015).

[24] J.H. Ziegeldorf, O.G. Morchon, K. Wehrle, Privacy in the internet of things: Threats and challenges, Secur. Comm. Netw. 7 (12) (2014) 2728–2742.

[25] A. Ukil, S. Bandyopadhyay, A. Pal, Iot-privacy: To be private or not to be private, in: Computer Communications Workshops (INFOCOM WKSHPS), 2014 IEEE Conference on, IEEE, 2014, pp. 123–124.

[26] X. Huang, R. Fu, B. Chen, T. Zhang, A. Roscoe, User interactive internet of things privacy preserved access control, in: Internet Technology And Secured Transactions, 2012 International Conference for, IEEE, 2012, pp. 597–602.

[27] A. Ukil, S. Bandyopadhyay, J. Joseph, V. Banahatti, S. Lodha, Negotiation-based privacy preservation scheme in internet of things platform, in: Proceedings of the First International Conference on Security of Internet of Things, ACM, 2012, pp. 75–84.

[28] J. Su, D. Cao, B. Zhao, X. Wang, I. You, ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the internet of things, Future Gener. Comput. Syst. 33 (2014) 11–18.

[29] L. Sweeney, k-Anonymity: A model for protecting privacy, Int. J. Uncertain. Fuzz. Knowl. Based Syst. 10 (05) (2002) 557–570.

[30] J. Domingo-Ferrer, V. Torra, A critique of k-anonymity and some of its enhancements, in: Availability, Reliability and Security, 2008. ARES 08. Third International Conference on, IEEE, 2008, pp. 990–993.

[31] N. Shen, J. Yang, K. Yuan, C. Fu, C. Jia, An efficient and privacy-preserving location sharing mechanism, Comput. Stand. Inter. (2015).

[32] B.H. Bloom, Space/time trade-offs in hash coding with allowable errors, Comm. ACM 13 (7) (1970) 422–426.

[33] A.H. Celdran, F.G. Clemente, M.G. Perez, G.M. Perez, Secoman: A semantic-aware policy framework for developing privacy-preserving and context-aware smart applications, IEEE Syst. J (2014).

[34] P. Gope, T. Hwang, Untraceable sensor movement in distributed iot infrastructure, Sensor J. IEEE PP (99) (2015), doi:10.1109/JSEN.2015.2441113. 1–1.

[35] P. Paillier, Public-key cryptosystems based on composite degree residuosity classes, in: Advances in CryptologyEUROCRYPT99, Springer, Berlin Heidelberg, 1999, pp. 223–238.

[36] J. Benaloh, Dense probabilistic encryption, in: Proceedings of the Workshop on Selected Areas of Cryptography, 1994, pp. 120–128.

[37] C. Gentry, S. Halevi, Implementing gentrys fully-homomorphic encryption scheme, in: Advances in Cryptology–EUROCRYPT 2011, Springer, Berlin Heidelberg, 2011, pp. 129–148.

[38] Z. Brakerski, V. Vaikuntanathan, Efficient fully homomorphic encryption from (standard) lwe, SIAM J. Comput. 43 (2) (2014) 831–871.

[39] J.-S. Coron, D. Naccache, M. Tibouchi, Public key compression and modulus switching for fully homomorphic encryption overthe integers, in: Advances in Cryptology–EUROCRYPT 2012, Springer, Berlin Heidelberg, 2012, pp. 446–464.

[40] J. Sen, Privacy preservation technologies in internet of things, in: Proceedings of the International Conference on Emerging Trends in Mathematics, Technology and Management, 2010, pp. 496–504.

[41] G. Sun, S. Huang, W. Bao, Y. Yang, Z. Wang, A privacy protection policy combined with privacy homomorphism in the internet of things, in: Computer Communication and Networks (ICCCN), 2014 23rd International Conference on, IEEE, 2014, pp. 1–6.

[42] R. Lu, X. Liang, X. Li, X. Lin, X.S. Shen, Eppa: An efficient and privacy-preserving aggregation scheme for secure smart grid communications, Parallel Distrib. Syst., IEEE Trans. 23 (9) (2012) 1621–1631.

[43] S. Sicari, L.A. Grieco, G. Boggia, A. Coen-Porisini, Dydap: A dynamic data aggregation scheme for privacy aware wireless sensor networks, J. Syst. Softw. 85 (1) (2012) 152–166.

[44] M.S.H. Talpur, M.Z.A. Bhuiyan, G. Wang, Shared–node iot network architecture with ubiquitous homomorphic encryption for healthcare monitoring, Int. J. Embedded Syst. 7 (1) (2014) 43–54.

[45] K.-S. Wong, M.H. Kim, Towards self-awareness privacy protection for internet of things data collection, J. Appl. Math. 2014 (2014).

[46] D. Chaum, E. Van Heyst, Group signatures, in: Advances in CryptologyEUROCRYPT91, Springer, Berlin Heidelberg, 1991, pp. 257–265.

[47] D. Boneh, X. Boyen, H. Shacham, Short group signatures, in: Advances in Cryptology–CRYPTO 2004, Springer, Berlin Heidelberg, 2004, pp. 41–55.

[48] C. Delerablée, D. Pointcheval, Dynamic fully anonymous short group signatures, in: Progress in Cryptology-VIETCRYPT 2006, Springer, 2006, pp. 193–210.

[49] X. Boyen, B. Waters, Full-domain subgroup hiding and constant-size group signatures, in: Public Key Cryptography–PKC 2007, Springer, Berlin Heidelberg, 2007, pp. 1–15.

[50] B. Libert, T. Peters, M. Yung, Group signatures with almost-for-free revocation, in: Advances in Cryptology–CRYPTO 2012, Springer, Berlin Heidelberg, 2012, pp. 571–589.

[51] K. El Defrawy, G. Tsudik, Prism: Privacy-friendly routing in suspicious manets (and vanets), in: Network Protocols, 2008. ICNP 2008. IEEE International Conference on, IEEE, 2008, pp. 258–267.

[52] A. Wasef, X. Shen, Efficient group signature scheme supporting batch verification for securing vehicular networks, in: Communications (ICC), 2010 IEEE International Conference on, IEEE, 2010, pp. 1–5.

[53] L. Malina, A. Vives-Guasch, J. Castellà-Roca, A. Viejo, J. Hajny, Efficient group signatures for privacy-preserving vehicular networks, Telecomm. Syst. 58 (4) (2015) 293–311.

[54] X. Zhu, S. Jiang, L. Wang, H. Li, Efficient privacy-preserving authentication for vehicular ad hoc networks, Veh. Technol. IEEE Trans. 63 (2) (2014) 907–919.

[55] F. Li, Z. Zheng, C. Jin, Secure and efficient data transmission in the internet of things, Telecomm. Syst. (2015) 1–12.

[56] J. Li, M.H. Au, W. Susilo, D. Xie, K. Ren, Attribute-based signature and its applications, in: Proceedings of the 5th ACM Symposium on Information, Computer and Communications Security, ACM, 2010, pp. 60–69.

[57] J. Camenisch, E. Van Herreweghen, Design and implementation of the idemix anonymous credential system, in: Proceedings of the 9th ACM conference on Computer and communications security, ACM, 2002, pp. 21–30.

[58] C. Paquin, G. Zaverucha, U-prove cryptographic specification v1. 1, Technical Report, Microsoft Technical Report, http://connect. microsoft. com/site1188, 2011.

[59] J. Hajny, L. Malina, Unlinkable attribute-based credentials with practical revocation on smart-cards, Springer, Berlin Heidelberg, 2013.

[60] A. Alcaide, E. Palomar, J. Montero-Castillo, A. Ribagorda, Anonymous authentication for privacy-preserving iot target-driven applications, Comput. Security 37 (2013) 111–123.

[61] X.-J. Lin, L. Sun, H. Qu, Insecurity of an anonymous authentication for privacy-preserving iot target-driven applications, Comput. Security 48 (2015) 142–149.

[62] J. Hajny, L. Malina, Z. Martinasek, O. Tethal, Performance evaluation of primitives for privacy-enhancing cryptography on current smart-cards and smart–phones, in: Data Privacy Management and Autonomous Spontaneous Security, Springer, Berlin Heidelberg, 2014, pp. 17–33.

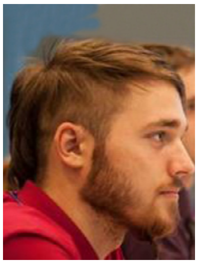
[63] M. Green, S. Hohenberger, B. Waters, Outsourcing the decryption of abe ciphertexts., in: USENIX Security Symposium, 2011, 2011.

[64] V. Goyal, O. Pandey, A. Sahai, B. Waters, Attribute-based encryption for fine–grained access control of encrypted data, in: Proceedings of the 13th ACM Conference on Computer and Communications Security, Acm, 2006, pp. 89–98.

[65] B. Waters, Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization, in: Public Key Cryptography–PKC 2011, Springer, Berlin Heidelberg, 2011, pp. 53–70.

Dr. Lukas Malina (malina@feec.vutbr.cz) is a researcher at the Department of Telecommunications at Brno University of Technology (BUT), Czech Republic. He accomplished his MSc. degree with honors and obtains the Dean prize for masters thesis at BUT in 2010. He received his Ph.D. degree from BUT in 2014. During his internship at Universitat Rovira i Virgili, Tarragona, Spain 2011-2012, he designed new group signature schemes with the batch verification. Currently, Lukas Malina deals with the privacy preserving cryptographic protocols, authentication schemes and developing the anonymous authentication systems on smartcards and smartphones. He also designs and develops lightweight cryptographic protocols for computational restricted devices. Further, he is interested in computer security and network security. He has published more than 40 papers in international journals and international and national conferences and is involved as a developer and researcher in several Czech scientific projects. He is a member of the Cryptology Research Group at BUT.

Dr. Jan Hajny (hajny@feec.vutbr.cz) has been with the Department of Telecommunications of FEEC BUT since 2008. Besides internship at the University of Minnesota, USA, from 2010 to 2011, he studied at the Department of Computer Science (Crypto-Group), the University of Arhus, Denmark in 2008. As a postdoctoral researcher, he deals with the design of privacy enhancing cryptographic protocols and authentication schemes. He is involved as a team leader in several industry-oriented projects, e.g. Cryptographic system for the protection of electronic identity, Integration server with cryptographic protection, as well as basic research projects, e.g. Research into cryptographic primitives for secure authentication and digital identity protection. Jan Hajny is a member of the Cryptology Research Group at BUT and IACR.



Radek Fujdiak (fujdiak@phd.feec.vutbr.cz) is a PhD candidate at the Brno University of Technology (Czech Republic). He received his MSc. degree in Communication Technologies at Brno University of Technology in 2013. He is involved in the research group of Doc. Jiri Misurec. His research activity is focused on issues related to the cyber-security, security, cryptography, in particular on low-power devices and Smart Technologies.

Dr. Jiri Hosek (hosek@feec.vutbr.cz) is a Senior Researcher in the Department of Telecommunications at Brno University of Technology, Czech Republic. He received his Ph.D. degree (2011) from Brno University of Technology. Jiri (co-)authored more than 60 published research works on networking technologies, wireless communications, quality of service, user experience assessment and machine-to-machine applications.