Ad Hoc Networks 32 (2015) 17–31

Contents lists available at ScienceDirect

Ad Hoc Networks

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

Survey on secure communication protocols for the Internet of Things

Kim Thuat Nguyen [a,*], Maryline Laurent [b,1], Nouha Oualha [a,2]

[a] CEA, LIST, Communicating Systems Laboratory, 91191 Gif-sur-Yvette CEDEX, France
[b] Institut Mines-Telecom, Telecom SudParis, UMR CNRS 5157 SAMOVAR, 9 rue Charles Fourier, 91011 Evry, France

ARTICLE INFO

Article history:
Received 5 June 2014
Received in revised form 20 November 2014
Accepted 11 January 2015
Available online 9 February 2015

Keywords:
Security
Key management
Internet of Things
Wireless sensor network
Cryptographic primitives

ABSTRACT

The Internet of Things or "IoT" defines a highly interconnected network of heterogeneous devices where all kinds of communications seem to be possible, even unauthorized ones. As a result, the security requirement for such network becomes critical whilst common standard Internet security protocols are recognized as unusable in this type of networks, particularly due to some classes of IoT devices with constrained resources. The document discusses the applicability and limitations of existing IP-based Internet security protocols and other security protocols used in wireless sensor networks, which are potentially suitable in the context of IoT. The analysis of these protocols is discussed based on a taxonomy focusing on the key distribution mechanism.

1. Introduction

The Internet of Things (IoT) is designed as a network of highly connected devices (things). In today perspective, the IoT includes various kinds of devices, e.g., sensors, actuators, RFID tags, smartphones or backend servers, which are very different in terms of size, capability and functionality. The main challenge is how to adapt such network so to operate in the conventional Internet. Inspired by that motivation, recent research efforts focus on the design, application and adaptation of standard Internet protocols in the IoT.

The initiative of 6LoWPAN [9] working group allowed the smallest devices with limited processing capabilities to become part of the Internet by enabling the use of IP over these devices. Such great feature enables the connection of literally billions of devices to the Internet, in which very different *things* such as a humidity sensor or an RFID tag can communicate with each other, with a human carrying a smartphone, or with a remote backend server.

While the concept of IoT is easy to grasp, major research efforts still need to be made. Various aspects of IoT are currently being discussed, such as IoT applications and architectures. In addition, more and more research efforts are initiated in resolving challenges associated with security, privacy, and trust as IoT devices are increasingly deployed. According to Gartner's forecast [21], the IoT, which excludes PCs, smartphones and tablets, will grow to more than 26 billion units installed in 2020. Allowing each single physical object to connect to the Internet and to share information, may create more threats than ever for our personal data and business secret information. Concerned objects cover our everyday friendly devices, such as, thermostats, fridges, ovens, washing machines, and TV sets. It is easy to imagine how bad it would be, if these devices were spying on us and revealing our personal information. As an example, a major cyber-attack campaign observed by

* Corresponding author. Tel.: +33 1 69 08 00 98.
E-mail addresses: kimthuat.nguyen@cea.fr (K.T. Nguyen), maryline.laurent@telecom-sudparis.eu (M. Laurent), nouha.oualha@cea.fr (N. Oualha).
[1] Tel.: +33 1 60 76 44 42.
[2] Tel.: +33 1 69 08 46 25.

http://dx.doi.org/10.1016/j.adhoc.2015.01.006
1570-8705/

Proofpoint's researchers [28] in January 2014, proved that even a harmless fridge can be employed to launch security attacks. Their analysis shows that 25 percent of malicious emails from the cyber-attack between December 23, 2013 and January 2014 (over 750,000 messages), came from "*smart*" things, including home appliances (TVs, refrigerators...). It would be even worse if critical IoT applications, for instance, the control system in nuclear reactors, the vehicle safety system or the remote monitoring in healthcare, were compromised.

By means of IP protocols crafted for the IoT, an IoT device is able to directly interact with other Internet entities located far beyond its local network. In a typical WSN, devices should be properly authenticated in the network based on a set of credentials stored in a secure area. The security solutions generally deployed within the network are poorly defined to protect communications within the network premises and not between external entities. To provide end-to-end security, the potential adaptations of several standard security protocols have been studied in [1] such as IKE/IPsec, TLS, DTLS, and HIP-DEX, but certain issues continue to persist using these solutions. In particular, resource limitations and the large volume of IoT devices deployed in a network hamper the application of Internet standard solutions.

According to the authors in [33], several new issues brought by IoT need also to be addressed, such as secure booting, firewalling and secure updating and patching. For example, we need to ensure that only authorized and authenticated software are loaded into the embedded device, for example, by verifying a digital signature attached to the software image. As stated in a recently HP security report [9], almost 60 percent of smart devices are not using encryption when downloading software updates. In order to deploy security solutions to this problem, devices are required not only to use cryptographic algorithms to perform encryption, but also to share the necessary keys required by these algorithms, which is an even worse issue considering the foreseen large deployment and the general resource limitations of these devices.

The main motivation of this survey is to identify security issues associated with IoT, and to demonstrate the limitations of existing security solutions to fulfill these issues. The reviewed solutions are analyzed and compared.

1.1. Related surveys and positioning

There have been several conducted studies and surveys [e.g., 60–64] that are relevant to the security in the IoT. For instance, Wang et al. [64] gave a very detailed survey of security issues in wireless sensor networks, which can be considered as a reference for the IoT. The authors identified the constraints and the requirements based on the existing attacks against the IoT at different layers. They also presented the key management systems in WSN according to the employed cryptographic primitives. Atzori et al. [61] focused on authentication, data integrity and privacy issues in the IoT, particularly in RFID systems and sensor networks. Kumar et al. [62] gave a general overview of security and privacy issues in IoT. They provided a description of different security threats and privacy concerns while processing, storing, and transmitting data. The main line of the existing surveys in relation with the IoT security is that they generally focus on identifying the challenges and the security threats present in the IoT. However, several security solutions and techniques have been proposed since the advent of the IoT. For this reason, the present survey takes a different direction by looking in depth into these security protocols and techniques. Indeed, we will not focus on specific security properties needed for the IoT. We will look closer at the security protocol itself, how it is constructed, which security properties are provided, and which cryptographic primitives are used. Moreover, the survey proposes a new taxonomy of key establishment mechanisms in the context of IoT that allows to better understand the proposed security approaches. In this way, strong and weak features of existing approaches can be identified with the objective to build secure protocols for the IoT.

The contributions of the document are threefold:

- present an overview of the challenges and the requirements to build a secure IoT;
- provide a taxonomy of different security protocols proposed for WSN and IoT with respect to the employed key bootstrapping mechanism and also propose a comparative analysis of these protocols and techniques; and
- finally, provide a review of ongoing research initiatives in the field of security in the IoT.

1.2. Paper outline

The rest of this paper is organized as follows. Section 2 discusses the security requirements and challenges associated with the IoT. Section 3 gives a classification of recently proposed security protocols for IoT. Sections 4 and 5 give in-depth description of the protocols based on asymmetric key schemes and the protocols based on symmetric key pre-distribution schemes. Section 6 evaluates the solutions according to the considered categories in terms of the challenges identified in Section 3. In Section 7, we look into promising security research directions for the IoT. Finally, concluding remarks are provided in Section 8.

2. IoT security overview

The IoT offers connectivity for both human-to-machine and machine-to-machine communications. In the near future, *everything* is likely to be equipped with small embedded devices which are able to connect to the Internet. Such ability is useful for various domains in our daily life: i.e. from building automation, smart city, and surveillance system to all wearable smart devices. However, the more the IoT devices are deployed, the greater our information system is at risk. Indeed, a non-negligible number of devices in IoT are vulnerable to security attacks, for example, denial of service and replay attacks, due to their constrained resources and the lack of protection methods. This kind of attacks leads to sensor battery depletion and results in poor performances of sensing applications. In more serious cases, information leak from such tiny

devices can expose sensitive data to the outside. In this section, we firstly present the essential security properties for the IoT. Then, we summarize the challenges to be addressed in the IoT.

2.1. Security properties

Several security properties may need to be satisfied in order to secure the IoT. These general security properties have been also identified in [1,44]. Generally, the security services that should be provided include confidentiality, integrity, authentication, authorization, and freshness. The security requirements are centered on data if sensitive data measured or shared by IoT devices may need to be protected. Security requirements may also involve controlled access to other resources, for instance the IoT network layer. Table 1 defines the security properties that will be discussed in this document in relation with the security protocols and solutions proposed for the IoT.

2.2. Challenges

The heterogonous nature of IoT raises various challenges in terms of data security and network functionality. A secure and operational IoT must overcome the challenges given in Table 2 in order to fulfill the above security requirements.

3. Taxonomy of security protocols for the IoT

The life cycle of a "*thing*" is composed of three phases (as denoted in [1]): bootstrapping, operational and maintenance phases. The bootstrapping phase refers to any processing tasks required before the network can operate. Sarikaya et al. [44] also define that this process involves a number of settings to be transferred between nodes that shared no prior knowledge of each other. The bootstrapping step of a device is complete when all security parameters (e.g., secret keys) are securely transferred to the device. This study focuses on recent security solutions proposed for a secure bootstrapping process. The terms and definitions used throughout the rest of the document are presented in Table 3.

In this section, we first describe the reference model that illustrates the scenario in which the considered security protocols can be deployed. We then present, in Section 3.2, our classification of the security protocols based on the key bootstrapping mechanism, and compare, in Section 3.3, our classification with related works.

3.1. Scenario under consideration

The security protocols analyzed in this document, as illustrated in Fig. 1, involve two entities. At least one of them is a device with resource constraints, whereas the second entity can be seen as another constrained device or an external Internet server (i.e., with rich resources). The considered network of "*things*" consists of a number of tiny nodes communicating with each other and with an unconstrained resource border router (6LBR). The 6LBR is the bridge between the sensor node and the outside world. The 6LBR may take part in the communication between two entities in a *passive* (transparent to the communicating parties) or *active* (as a mediator in the communication process) manners. Our study concentrates mainly on securing unicast communications between two entities. Note that group communications are out of scope of this document.

3.2. Classification

In this document, existing security solutions for IoT is categorized into two main types: solutions that rely on asymmetric key schemes and solutions that pre-distribute symmetric keys to bootstrap a secure communication. This section describes the two first levels of the proposed taxonomy.

- **Asymmetric key schemes (AKSs):** The key schemes based on asymmetric cryptography, also known as Public-key cryptography (PKC) are considered as a very common approach to establish a secure communication between two (or more) parties. They employ asymmetric algorithms and are widely deployed in the conventional Internet. The applicability of AKSs in the IoT has one major inconvenience, which is the computation cost and energy consumption. Despite of expensive operations, a lot of researches still seek to apply AKSs in the context of IoT. The proposed approaches can be classified into two categories: key transport based on public key encryption and key agreement based on asymmetric techniques.
  - *Key transport based on public key encryption:* Similarly to the traditional key transport mechanism, the first category requires from the public key to securely transport information. Various key establishment techniques have been proposed for IoT, ranging from raw public key usage to complex implementations in X.509 standard.
  - *Key agreement based on asymmetric techniques:* The second category is based on asymmetric primitives in which a shared secret is derived among two or more parties. In this category, we notice obviously the DH protocol [11] and its variants as we will mention later.
- **Symmetric key pre-distribution schemes:** In addition to asymmetric approaches, researchers propose also multiple techniques using symmetric key establishment mechanisms to bootstrap secure communication in the IoT. Symmetric approaches often assume that nodes involved in the key establishment share common credentials. The pre-shared credentials might be a symmetric key or some random bytes flashed into the sensor before its deployment. This category can be divided into two main sub-categories:
  - *Probabilistic key distribution:* This sub-category concerns the mechanisms that distribute security credentials (keys, random bytes) chosen randomly from a key pool to constrained nodes. During their initial communication, each two nodes may discover a common key, with certain probability, to establish a secure communication.

Table 1
Security properties for security protocols in IoT.

Confidentiality	Exchanged messages in the IoT may need to be protected. An attacker should not gain knowledge about the messages exchanged between a sensor node and any other Internet entity
Integrity	The alteration of messages should be detected by the receiver
Authentication	The receiver should be able also to verify the origin of the exchanged messages
Authorization	IoT devices should be able to verify whether certain entities are authorized to access their measured data. At the network layer, only authorized devices should be able to access the IoT network. Unauthorized devices should not be able to route their messages over the IoT devices, because it may deplete their energy
Freshness	This property ensures that no older messages are replayed. This is important to secure the communication channel against replay attacks

- *Deterministic key distribution:* In this sub-category, a deterministic design is applied to create the key pool and to distribute uniformly the keys such that each two nodes share a common key.

Fig. 2 summarizes our taxonomy. Each class of the security solutions provides its own advantages and disadvantages, as it will be discussed in Sections 5 and 6.

3.3. Related work in IoT security protocol classification

Classification approaches have been proposed in several works [10,52,63,64]. In [10], the authors propose several ways to classify key establishment approaches, for instance based on the employed authentication method or the underlying cryptographic primitive. Camtepe and Yener [52] give a detailed classification of symmetric key distribution protocols for two different scenarios: distributed and hierarchical WSNs. In each scenario, the authors analyze diverse mechanisms to establish pair-wise and group-wise keys between sensor nodes. Similarly, Wang et al. [64] propose a classification of symmetric key management protocols in WSN, but based on the network structure and the probability of key sharing between a pair of sensor nodes. Their works at a very first level differentiate centralized and distributed key schemes. At a second level, they provide other differentiation based on the probabilistic and deterministic key establishment mechanisms. Roman et al. [63] give a high level classification based on the key management systems (KMS), namely: key pool framework, mathematical framework, negotiation framework and public key framework. They conclude that public key cryptography can be a viable solution for sensor nodes that run as client nodes (in the model client–server). For server nodes, mathematical-based KMS, such as polynomial scheme, provide better performances. The aforementioned approaches do not sufficiently cover possible key distribution mechanisms (asymmetric and symmetric methods), for example, only symmetric approaches are studied in [52,64]. Besides, they provide heterogeneous classifications due to unrelated different criteria, as in [63,64].

By taking into account the classifications described above, especially in [10], our taxonomy covers asymmetric key distribution mechanisms for IoT, in addition to symmetric approaches. The taxonomy is marking out different protocols by the key establishment scheme used to establish a secret session key: asymmetric or symmetric techniques. As mentioned in Section 3.1, we do not consider protocols that establish group-wise keys between sensor nodes for which interested readers could refer to [52]. Only pair-wise key establishments are considered in this paper. Our taxonomy has a high classification degree leading to a more in depth protocol evaluation. For instance, in the asymmetric approach, we do not only discuss on the applicability of public key cryptography in the context of IoT, as described in [63], but we also differentiate different asymmetric key schemes based on the key delivery scheme (key transport or key agreement). In symmetric key pre-distribution schemes, we organize the existing security

Table 2
Research challenges in IoT.

Interoperability	Deploying security solutions in the IoT should not hinder the functional operation of interconnected heterogeneous devices
Resource constraints	Most of IoT devices are limited in terms of CPU, memory capacity and battery supply. They often operate on lossy and low-bandwidth communication channels. It seems to be impossible to apply directly standard conventional security protocols of the Internet in the context of IoT. As an example, the use of small packets (i.e. IEEE 802.15.4 supports only 127-bytes packets [25]) may result in fragmentation of larger packets when using the standard protocols. This will exhaust the life time of sensor nodes and open new possibility of DoS attacks. Hence, the standard security protocols must be redesigned to adapt such difficult scenario, in order to offer equivalent security levels but more efficient performance for the IoT
Availability	The sensor nodes must be available when needed. High availability network of things should remain functional, especially against denial-of-service attacks, such as flooding of incoming messages to targeted nodes forcing them to shut down
Resilience to attacks	The system has to avoid single points of failure so a compromised node will not affect the whole system. Besides, the secured network must also avoid the resource-depletion attacks launched against resource-constrained devices
Privacy protection	The popularity of RFID tags has raised privacy concerns because anyone can track tags and find the identity of the objects carrying them. In addition, as wearable technology increases its pace, we will be soon able to connect our bodies to the Internet by "putting on" tiny hardware devices (ex. Implant chips inside our bodies). Consequently, our personal information (i.e. healthcare records) must remain secured and should not be traceable, linkable and identifiable
Scalability	The IoT network, for instance WSNs, is generally composed of a large number of devices. The proposed security protocol should be able to scale. This property is tightly related to the amount of information that each device has to keep in memory for a secure channel to be negotiated with as many entities as possible (other sensor nodes or Internet entities)

Table 3
Abbreviations and notations.

Abbreviation	Definition
IoT	Internet of Things
WSN	Wireless Sensor Network
PKC	Public Key Cryptography
KDC	Key Distribution Center
6LBR	6LoWPAN Border Router
PKG	Private Key Generator
DH	Diffie–Hellman exchange
IBE	Identity-based Encryption
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie–Hellman exchange

protocols into two categories: probabilistic and deterministic key distribution. These categories have also been mentioned in [52,64]. However, in the deterministic approach, we go further by distinguishing protocols that have server(s) participating in the key negotiation process from protocols that do not depend on any third party during key establishment phase.

4. Asymmetric key schemes

The position of asymmetric cryptography or PKC is clear in the conventional Internet. However, it is not the case in the context of IoT because of its expensive encryption and verification operations. However, the development and implementation of PKC in IoT has never been stopped. In fact, new improvements of several primitives (i.e. ECC, NTRU) continue to reduce the cost of cryptographic operations, so the PKC approach is of a growing interest for constrained environments. A brief study in the following sections demonstrates various possible forms of asymmetric key schemes in IoT.

4.1. Key transport based on public key encryption

This sub-category looks into the key establishment schemes where the public key is used to transport secret data or to negotiate a session key. Several methods are used to generate the pair of public and private keys. In this sub-category, we classify these mechanisms based on the public/private keys generation methods.

Fig. 3 gives an example of a communication scenario between two entities A and B. In this scenario, A and B can use directly the public keys to create an encrypted channel. The Certificate Authority (CA) may participate to verify the identity of the message transmitter when certificates are supported. This method can be expensive for resource-constrained-sensor nodes, in particular when using a traditional algorithm like RSA. Without a verifiable relationship between the public key and the identity (i.e. ID-based cryptography, cryptographic-based ID or with CA mediation), this approach becomes vulnerable to the man-in-the-middle attack. Indeed, both A and B cannot authenticate each other's identity. An attacker may generate any public/private keys and pretend to be A when communicating with B.

4.1.1. Raw public key encryption

Some mechanisms assume that the public key has been distributed beforehand or using out-of-band communications. These mechanisms offer small number of message exchanges but they are not scalable, because the public keys of all devices should be known by each device.

Some "*raw public key encryption*" mechanisms, i.e. Rabin's scheme [19] or NtruEncrypt [27] have been recommended for WSNs.

Rabin's scheme is very similar to the RSA algorithm (widely used cryptosystem), which is also based upon the hardness of the factorization problem. In fact, the scheme requires the same energy consumption for decryption operations than RSA with the same security level. However, it offers much faster mechanism for encryption operations because only one squaring is required to encrypt a message.

NtruEncrypt is a cryptosystem which is known to be a lattice-based alternative to RSA and ECC (Elliptic Curve Cryptography) primitives. The mechanism is highly efficient and suitable for the most limited-resource devices such as smartcards and RFID tags. In [27], the authors give a comparison of the three PKC mechanisms proposed for

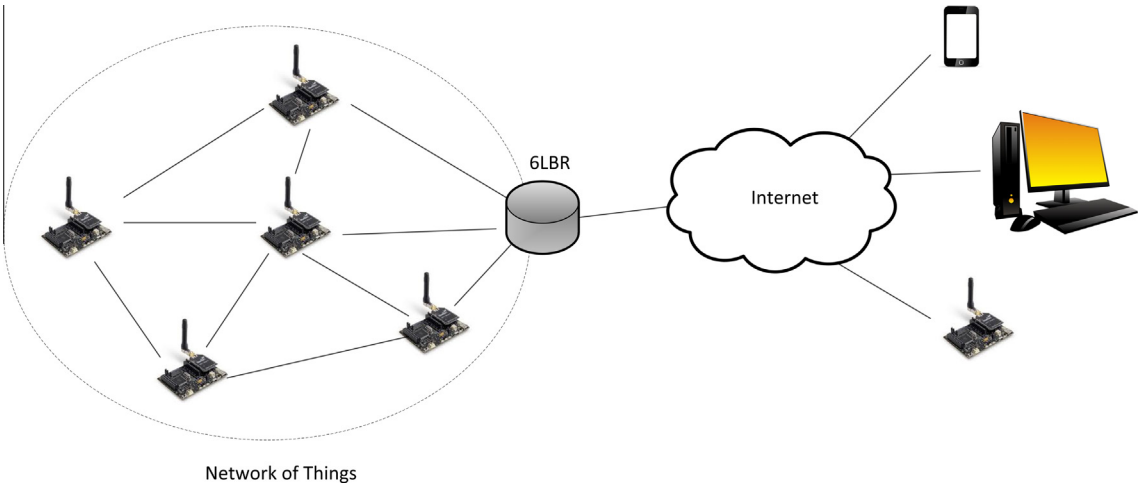

Fig. 1. Network architecture of our scenario.

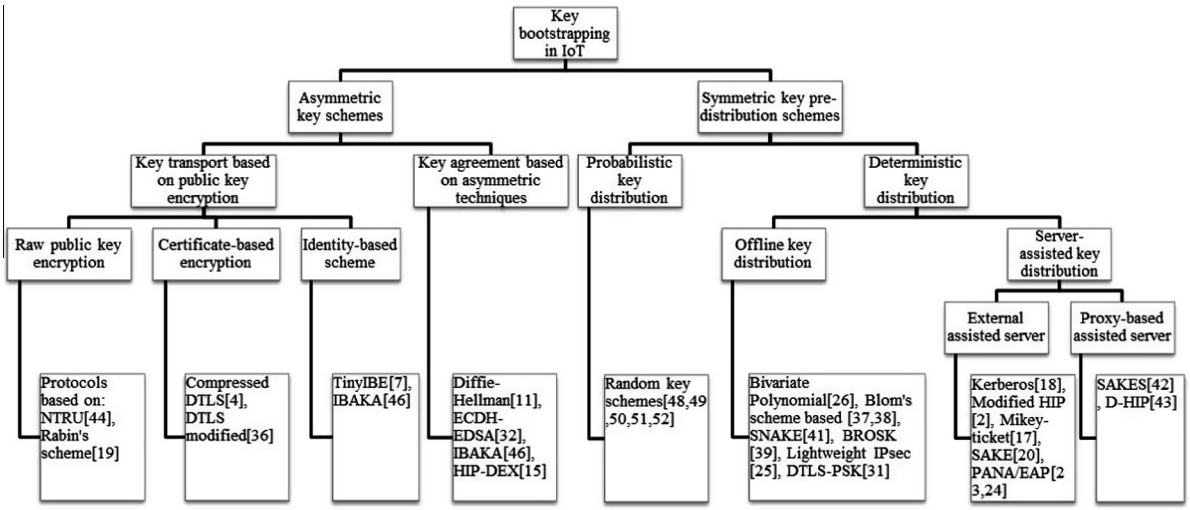

Fig. 2. Classification of key bootstrapping mechanisms in IoT.

constrained devices: the Rabin's scheme, NtruEncrypt and ECC. The results show that NtruEncrypt leads to the smallest average power consumption. Nevertheless, this cryptosystem often requires large-size messages, and might result in packet fragmentation at lower layers and many re-transmissions in the presence of communication errors.

The protocols that are based on *"raw public key encryption"* require small number of exchanged messages; this is actually advantageous if the transmission power is the most important and limiting factor.

4.1.2. Certificate-based encryption

Certificate-based protocols are a popular choice to establish a secure communication between two entities over Internet. The trust relationship between the two entities is guaranteed by a well-known third party (CA) using the standard X.509 certificate that validates the identity of the entity as illustrated in Fig. 3. Indeed, each sensor node possesses a certificate signed by the trusted CA. This latter can be loaded into the node before the deployment or can be directly acquired on request from a trusted party.

TLS [12] has been recommended by many standards specified by IETF (Internet Engineering Task Force) for security services. However, it is mentioned in [3,4] that TLS is not a wise choice with respect to the security best practices in IoT. In fact, TLS runs normally in a reliable transport protocol like TCP which is unsuitable for constrained resource devices, due to its congestion control algorithm. As a replacement for TLS in the tightly constrained environments, the DTLS (Datagram Transport Layer Security) protocol has been proposed recently. It operates over the unreliable transport protocol i.e., UDP and provides the same high security levels as TLS.

The utilization of a certificate is basically expensive. To reduce the power consumption, both hardware and software related improvements have been considered by researchers:

Usage of cryptographic hardware accelerators: The hardware accelerators are in charge of all cryptographic computations. Kothmayr et al. [3] propose a method to implement DTLS using hardware assistance on sensor nodes. The solution assumes that each sensor is equipped with a TPM (Trusted Platform Module). A TPM is an embedded chip that offers secure generation of cryptographic keys and sealed storage as well as hardware support for cryptographic algorithms. The fully authenticated handshake can be performed between a sensor (equipped with TPM) and a subscriber (another sensor or external entity). Both sensor and subscriber transmit their X.509 certificate to initiate the authentication phase. These certificates are signed by a trusted CA and are included in a fully authenticated DTLS handshake. This solution not only has a high security level by establishing the trusted relationship with the assistance of an approved third party, but it also provides message integrity, confidentiality and authenticity with affordable energy, end-to-end latency and memory overhead as claimed by the authors.

Nevertheless, the approach is expensive and complex with respect to deploying a hardware accelerator next to every sensor, especially for large number of sensors.

Optimization of existing protocols (software implementation): A security protocol employing the certificates is tailored to provide higher performance without affecting the robustness of the protocol. Raza et al. [4] propose a modification of DTLS using the 6LoWPAN compression mechanism [14]. The modified protocol reduces the size of some headers (i.e. the DTLS record header, the handshake header, the handshake message). These changes improve the performance of DTLS in terms of packet size, energy consumption, processing time and network response time. However, the proposed solution does not propose backward compatibility with the actual DTLS standard, in particular with respect to header compression.

Hummen et al. [36] propose a design idea to effectively reduce the overhead of the DTLS handshake. Full handshake procedure requires 15 message exchanges, high dynamic storage capability (RAM) during the communication and long processing time for cryptographic tasks. In order to mitigate the full handshake inconvenience, the authors propose to delegate the handshake procedure to a rich-resource

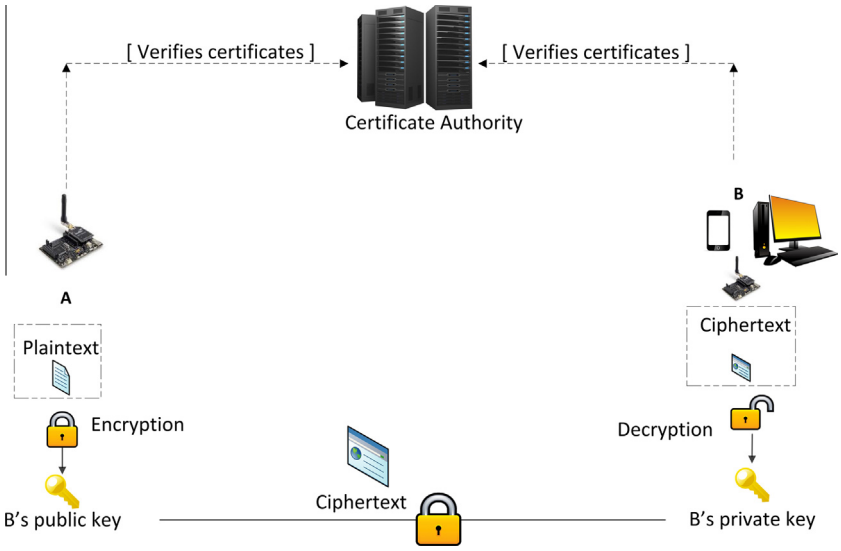

Fig. 3. Public key transport mechanism.

entity, e.g. the gateway or the device's owner. All certificate related tasks are performed in the rich-resource entity and only the session-state message is sent to the constrained device. The session can then be established using this message with no additional calculation. This modified DTLS can highly reduce communication overhead at the condition that the rich-resource server is trusted.

Granjal et al. [31] present similar modifications to DTLS, but the DTLS handshake is mediated by the 6LoWPAN Border Router (6LBR). The 6LBR participates in the secure communication but is transparent to sensing devices and the Internet host. The border router intercepts and forwards packets at the transport-layer. From the point of view of the Internet host, it communicates with the 6LBR using traditional DTLS protocol where authentication is supported by ECC based certificate. On the other side, the 6LBR operates in the pre-shared key security mode for communicating to the constrained sensing devices. Moreover, the 6LBR authenticates the nodes with a mechanism inspired by Kerberos [18]. If the authentication is successful, a secret session key is generated to secure the communication between the sensing devices and the 6LBR. Actually, it is used to encrypt the pre-master key in the *ClientKeyExchange* message that the Internet host sends to the 6LBR. When the pre-master secret key is computed successfully in the Internet host and the sensing device, end-to-end DTLS security is enabled. The proposed architecture delegates all the expensive operations (ECC computation, key agreement...) to the border router so that it offers better lifetime for sensing devices. Nevertheless, the 6LBR is considered to be a single point of failure.

The IKE protocol [59] works usually jointly with IPsec to provide security associations (SAs) between two entities. This protocol has a variant where the mutual authentication is enabled using RSA-based certificates. Ray and Biswas [59] propose another variant for IKE that is based on ECC-based public key certificate for authentication and ECDH for key agreement instead of RSA and DH protocol. The proposal reduces the computation cost as it is mainly limited to the point multiplication operations and it requires smaller key size than RSA for the same level [30].

4.1.3. Identity-based schemes (IBS)

The first implementation of Identity-Based Cryptography was developed by Shamir [6]. This type of cryptography defines a well-known string (identity) representing an individual or an organization, which is used as a public key. The private key of each entity is generated from its public key by a trusted party (Fig. 4), named a Public Key Generator (PKG). This solution eliminates the need for certificates, which makes the solution advantageous especially for WSNs. Indeed, any sensor nodes can simply generate the public key of other nodes when needed to establish a secure communication using their identities. In addition, the revocation mechanism is supported by consulting the list of valid sensor identities. However, ID-based schemes are vulnerable to key-escrow attacks as the PKG knows the private keys of all nodes in the network. It can impersonate any node and consequently intercept all the traffic in the system. Therefore, the PKG is always considered as well protected and trusted by all network nodes.

In a constrained environment, IBE paradigm is mostly implemented using the ECC primitive [46,31]. Implementations on other primitive exist, for example, RSA or ElGamal-type IBE [47]. Nevertheless, they are too much expensive for constrained nodes because they are based on exponentiation operations with a large exponent. Yang et al. [46] propose IBAKA – an IBE scheme inspired by Boneh et al. scheme [45]. However, they tailor the IBE method into an ECDH [32] key exchange in order to establish a session key. Their proposal still requires 2 bilinear pairings and 3 scalar point multiplications each time a secret key is bootstrapped.

Szczechowiak and Collier [7] propose TinyIBE – a very simple authenticated key distribution based on IBE for heterogeneous sensor networks. The scheme requires no pairing calculation. It is able to retrieve a session key for two nodes after only 2 message exchanges.

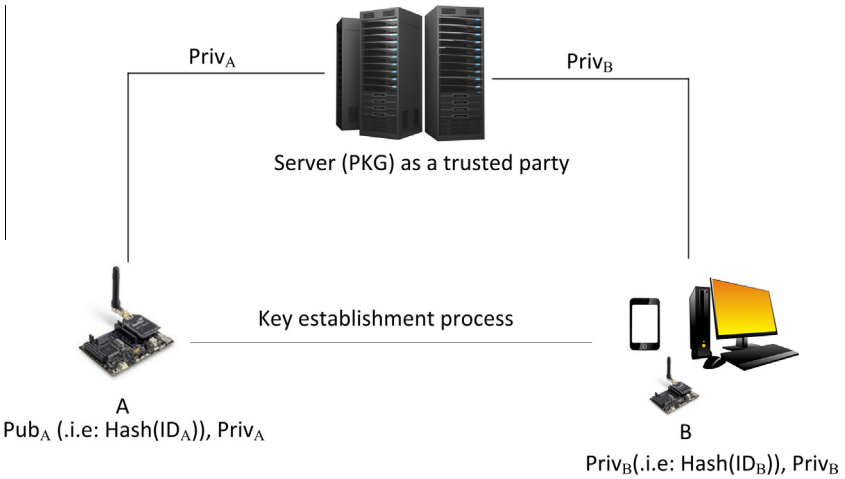

Fig. 4. Identity-based cryptography infrastructure.

4.2. Key agreement based on asymmetric techniques

This sub-category is about key agreement protocols based on asymmetric primitives in the IoT. As mentioned in various research works, a key agreement protocol is the mechanism where two (or more) parties derive a shared secret and no other party can predetermine the secret value. Fig. 5 illustrates the process of a typical asymmetric key agreement. Km is the secret generated after the agreement procedure. This symmetric key is then used to secure the communication.

The Diffie–Hellman (DH) protocol [11] and its variants are classical examples for symmetric key agreement. However, DH protocols are considered expensive and unsuitable for the constrained nodes in particular, for class 0 and 1 according to the node classification in terms of resource capacity in lwig-terminology [29].

Some variants of the DH protocol are considered in constrained environments using ECC, i.e. ECDH. The ECDH cryptographic primitive offers smaller key size than RSA. Indeed, the US National Institute for Standard and Technology (NIST) in [30] has showed that to achieve the security level of 128-bit AES key size, one can prefer 256 bit key size using elliptic curve instead of 3072 bit parameters in RSA and DH protocol. As an example, de Meulenaer et al. [32] implemented a key agreement protocol based on ECDH providing authentication using the Elliptic Curve Digital Signature Algorithm (ECDH-ECDSA).

Practical measurements on the MICAz and the TelosB sensors showed that ECDH-ECDSA is affordable in terms of computation complexity.

IBAKA [46] proposes a combination of ECDH and IBE for sensor networks. The scheme relies on the ECDH protocol, and additionally provides the privacy of message exchanges using Boneh et al. identity-based scheme [45].

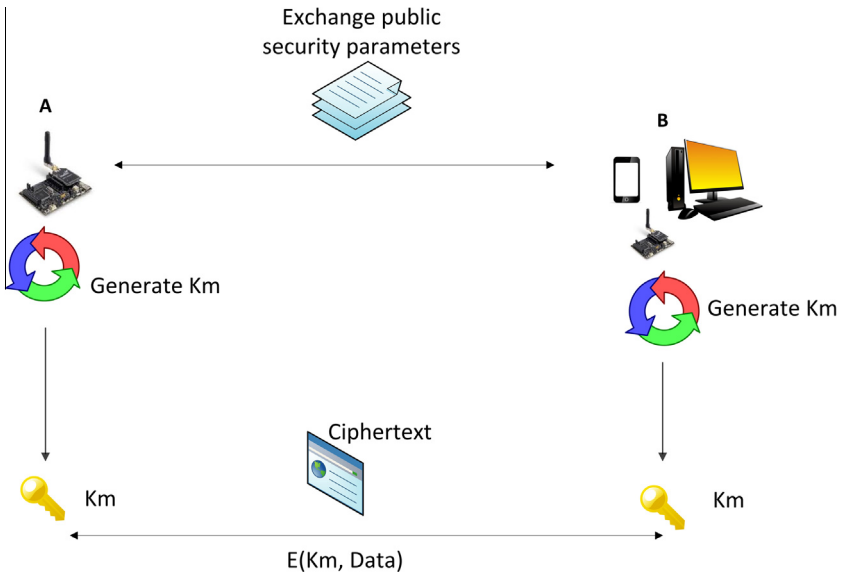

Fig. 5. Key agreement based on asymmetric mechanisms.

HIP-DEX (Host Identity Protocol Diet Exchange) [15] applies also the DH protocol to generate a session key between two entities after only a 4-messages exchange. This protocol is a variant of HIP Base Exchange [53] specially designed to reduce the complexity of cryptographic computations. It uses the smallest possible set of cryptographic primitives (e.g., AES-CBC instead of cryptographic hash functions), removes digital signatures and implements static ECDH to encrypt the session key, etc. This protocol has been largely taken into consideration in the context of IoT by many recent works [2,53]. For instance, Meca et al. [2] propose an efficient network access mechanism based on HIP-DEX for mobile nodes joining the local sensor network. Besides, Hummen et al. [53] tailor HIP-DEX to the IoT, in particular, by adapting the session resumption mechanism as in TLS [13]. As such, the constrained node performs expensive operations once and maintains session-state for re-authentication and re-establishment of a secure channel.

The key agreement protocols based on DH require fewer messages to establish a session key but the computational tasks on sensor nodes are usually complex.

5. Symmetric key pre-distribution schemes

In this sub-category, the communicating parties often initially share some credentials before bootstrapping the communication. The key pre-distribution mechanisms may differ as described in the following sections.

5.1. Probabilistic key distribution

The mechanism of random key pre-distribution (RKP) was first proposed by Eschenauer et al [48]. A typical RKP consists of three phases: key pre-distribution, shared-key discovery and path-key establishment. In the scheme, a large key pool is generated. Keys are then randomly selected from the key pool and distributed to sensor nodes. Any two nodes may share a common key with a certain probability. The third phase is triggered when two nodes do not share any common key. In this process, one node first generates a random key K. It then sends the key to its neighbors using the pre-established secure channel. The process continues until the key K arrives at the other node. K is considered afterward as the pairwise key between both nodes.

Several solutions are inspired by this scheme [37,49–51]. These proposals improve specially the pre-distribution phase to enhance the key connectivity between nodes and reduce the memory space needed for key storage. In fact, Du et al. [37] propose a key pre-distribution scheme that relies on the deployment knowledge and avoids unnecessary key assignments. Ito et al. [50] develop a scheme based on Du et al. [37] works but the keys are mapped on two-dimensional positions. They propose a probability density function which provides better key connectivity. Chan et al. [49] develop also a mechanism to reinforce the path-key establishment phase. The basic idea is that node A finds all possible links to a node B. It generates for each link a random value and routes these values to B. The common keys between A and B are protected by these random values. The generated key will be shared by both nodes, unless the adversary manages to eavesdrop on all paths between them.

The *probabilistic key distribution* generally does not guarantee session key establishment between all nodes even with the path-key establishment phase. Two nodes may not share any common keys with a certain probability.

5.2. Deterministic key distribution

In this sub-category, the described key schemes rely on a deterministic process to generate the key pool and to distribute keys to nodes in order to guarantee secure full connectivity in the network. In deterministic solutions, the key schemes are distinguished by the presence or not of a trusted third party during the key bootstrapping.

5.2.1. Offline key distribution

The offline key distribution method is widely used in WSNs because of its simplicity. Depending on the used protocol, every node in the same network may share a network key or each two nodes may have a common pairwise key. The session key is then generated after very few data exchanges without the presence of any third party. The offline key distribution provides efficiency in terms of energy consumption because it does not require expensive cryptographic computations like asymmetric approaches. However, when a sensor node is physically attacked, the secret data stored inside the node can be exposed. Consequently, the attacker can gain access to several nodes which share the secret key with the attacked node, or in the worst case, it may access the whole network.

In several existing works, mathematical properties have been applied to create the model for securing key exchanges between sensor nodes. These mechanisms are still applicable in the context of IoT. The most well-known schemes are based on bivariate polynomials [8,26]. In these schemes, a node A shares with other nodes a bivariate n-degree polynomial $f(x,y)$. A can obtain the pairwise key with another node B by calculating the value of $f(Id_A, Id_B)$, where Id_A and Id_B are the respective identities of A and B. In the same way, B can obtain the same pairwise key, since $f(Id_A, Id_B)$ is equal to $f(Id_B, Id_A)$. In another scheme, called the Bloom's scheme [38], a secret symmetric matrix D is generated from the shared secret key between two nodes A and B. Each of them generates a public matrix I_A and I_B respectively for A and B. The private keys are respectively $priv_A = D \times I_A$ and $priv_B = D \times I_B$ for A and B. Finally, the pairwise key is calculated by solving $(priv_A \times I_B)$ or $(priv_B \times I_A)$. The problem with these latter two schemes is that the session key will remain unchanged for every two nodes.

SNAKE [41,39]and BROSK [40] are two key establishment schemes where the session key is generated without the need for a key server to perform key management. These two protocols assume that all nodes in the same network share a master secret key. In SNAKE, the session key is obtained by hashing two random nonces generated from each communicating party using the pre-shared key. BROSK broadcasts the key negotiation message containing a nonce. Once a node receives the message from its neighbors, it can construct the session key by computing the message authentication code (MAC) of two nonces.

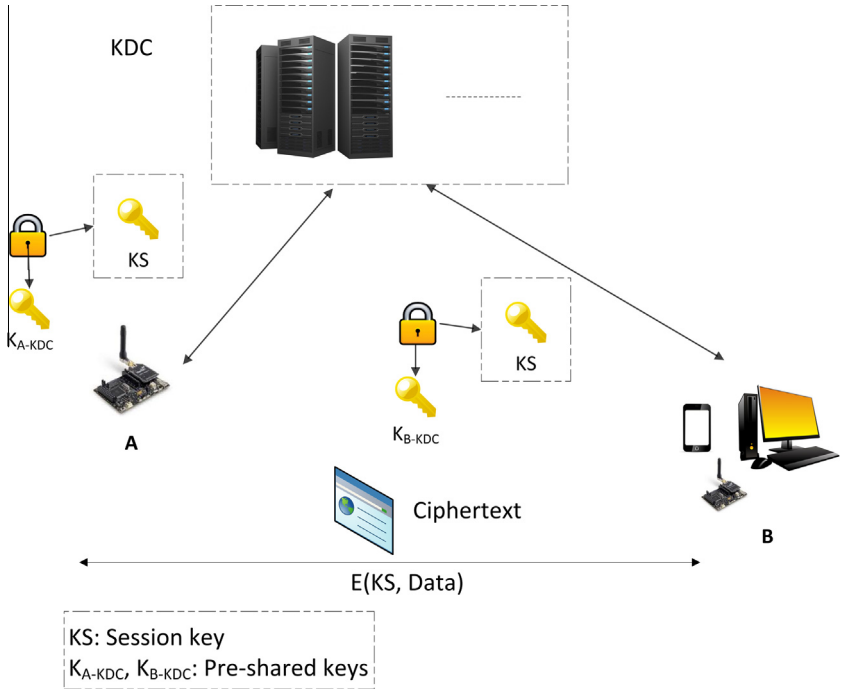

Fig. 6. Server-assisted mechanism.

Raza et al. [25] implement the standard Internet security protocol IPsec in an IP-based WSN (using 6LoWPAN). The authors propose mechanisms to compress the AH and ESP header in order to integrate IPsec with the 6LoWPAN layer but they keep a reasonable packet size. AH and ESP mechanisms provide origin authenticity, message integrity and confidentiality protection of IP packets but they do not handle the key exchange. The security associations are established manually using pre-shared key.

The *offline key distribution* does not provide rekeying operations. When the system changes to other secret keys, all the entities in the network need to be updated to establish secure communications using the new keys.

5.2.2. Server-assisted key distribution

Due to the resource limitations of constrained devices, the cryptographic computation and other expensive tasks (e.g., identity management, key generation) can be handled at rich–resource servers. Server-assisted approaches for key establishment protocols have been proposed in this respect in IoT. In such protocols, message exchanges engage two entities and one (or more) trusted server. The server shares long-term key *a priori* with each communicating entity. It often plays the role of a Key Distribution Center (KDC) and then supplies the session key to each party by re-encrypting it using the shared keys as shown in Fig. 6.

(1) External assisted server

In this sub-category, the assisted entities are external rich-resource servers which are located outside the WSN. As a result, they can handle the key distribution of one or several WSNs.

The second approach proposes in [3] is inspired by the TLS Pre-Shared Key cipher suite [13]. Each sensor has to

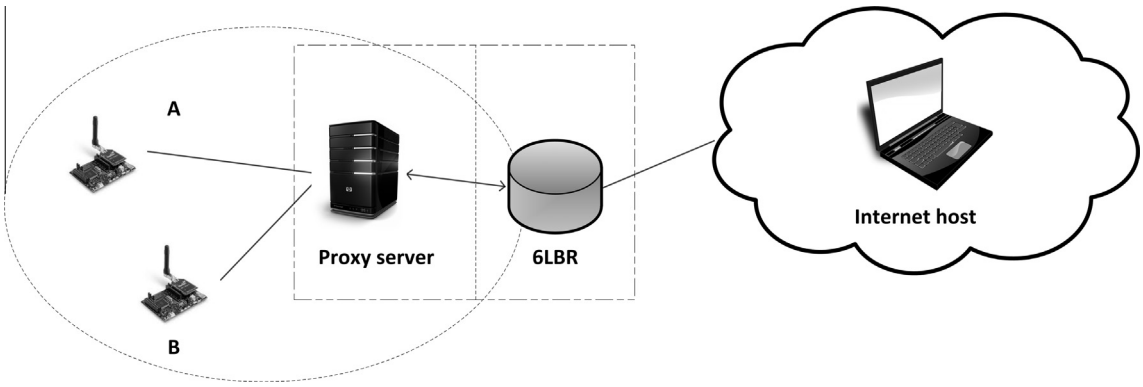

Fig. 7. Proxy-based assisted server infrastructure.

pre-install some random bytes called protokeys before the deployment. These random bytes are used to derive the PSK (Pre-Shared Key) key for each session. Instead of using TPM, a central rich-resource server is employed to create the security association between the sensor node and the subscriber. The protokeys are also known by the trusted server. The server then generates the same session key for the subscriber from the protokeys.

MIKEY-Ticket [17] is an additional mode to the basic MIKEY [22] protocol, in which a KDC is involved in the process of establishing a security association between the two parties. MIKEY-Ticket originated from the ticket concept of Kerberos [18]. The KDC securely communicates with the node initiating the protocol (Initiator) and the responding node (Responder) by encrypting important data using the pre-shared master key shared with each node. Nevertheless, the protocol is vulnerable to Denial of Service (DoS) attacks, particularly replaying messages to the Responder. To prevent these attacks, Boudguiga et al. [20] propose a new key establishment, called SAKE (Sever Assisted Key Establishment) based on the MIKEY-Ticket mode but removing the threat of DoS attacks. SAKE allows establishing security associations between the two parties after only five exchanged messages, compared to six messages in the original MIKEY-Ticket. Indeed, upon reception of the first message from the Initiator, the KDC generates the session key and contacts directly the Responder. This change reduces one message exchange comparing to MIKEY-ticket. Besides, as each message of SAKE contains a MAC computed with a key shared with the receiver, the DoS attack is mitigated.

Other IoT solutions of key distribution based on an external server, include solutions that implement the PANA protocol (Protocol for Carrying for Network access) [23]. PANA runs over UDP and uses EAP [5] (Extensible Authentication Protocol) for authentication that supports multiple authentication methods including pre-shared key distribution. Kanda et al. [24] propose an improvement of PANA to adapt the resource-constraints. The main modifications consist of reducing the number of message exchanges (e.g., choosing EAP-PSK as the only authentication method), removing unused PANA header fields, minimizing the collection of cryptographic primitives at the constrained device. These proposals may effectively reduce the PANA implementation code size at the device, but the authors do not give an estimation of the gains that might be obtained, for example, in terms of energy consumption or network-response time.

(2) Proxy-based assisted server

This sub-category does not require an external server but a proxy-based server (PBS) located within the WSN, as shown in Fig. 7. This server is equipped with sufficient resources and storage capacity to execute all expensive tasks for constrained nodes. It often plays the role of a mediator to associate the sensor nodes and other entities. Additionally, the PBSs usually share a symmetric secret key with the constrained nodes and the 6LBR router.

Using the same considerations, Hussen et al. [42] propose SAKES providing secure authentication and key establishment between a sensor node and an external Internet host. Upon the reception of a sensor node request, the PBS authenticates the sensor node with the help of 6LBR. It then applies a DH key agreement mechanism with the remote server and calculates the session key (SK) on behalf of the sensor node as the sensor node is resource constrained. Finally, the sensor node can communicate with the remote server in a secure manner using the SK received from the PBS.

In this same sub-category, Saied and Olivereau [43] present the Distributed HIP Exchange (D-HIP) protocol inspired by HIP-BEX [54]. They use the same network model as described in Fig. 7. During the key negotiation step, a constrained node establishes a session key with the server using the DH protocol by delegating the 2 modular exponentiation operations to the proxy nodes. It first splits its secret exponent a into n parts $a_1, a_2, \ldots, a_n$ where n is the number of the less constrained nodes. It then sends each part a_i to a neighbor node (proxies) PBS_i. The node PBS_i calculates its part of the final DH session key: $SK_i = (g^b \bmod p)^{ai}$ where the value $(g^b \bmod p)$ is achieved from the remote server (or Internet host). PBS_i sends SK_i back to the constrained node. From these values, the constrained node obtains the same final DH session key as the server (by multiplying the n values received). This approach has a major advantage that all expensive computation tasks are done by the PBS nodes. However, the number of message exchanges can be large depending on the number of PBS nodes. As we know, the transmission cost is non-negligible and packet lost during communication can happen at any time.

6. Discussion

Table 4 illustrates examples of security protocol solutions which are implemented in WSN and IoT. It compares these solutions using the identified criteria given in Section 2.

At first glance, we can easily identify that most of the general security services are well provided by the proposed protocols. Nevertheless, few protocols support the *Access control* (AC) and *Privacy Protection* (PP) properties. The AC service is very important and needed in such perspective where an Internet host can only access the sensor node to execute actions or to retrieve data according to its access privileges. The server-based protocols usually offer this requirement, for example, with the help of an authorization server. On the other hand, the PP strengthens the anonymity of communications. This property becomes very important in today perspective as personal data on sensor nodes must remain untraceable by any attackers.

In the high level synthetic picture, the table shows that the asymmetric solutions usually require high computation complexity on sensor nodes. However, these approaches have high resilience against node capture attacks, low memory requirements for keying materials, few message exchanges and high scalability for large networks. On the other hand, the key pre-distribution schemes offer low complexity computation which is really beneficial for constrained nodes, but, they have their own inconveniences, such as high communication complexity,

Table 4
Summary of proposed security solutions for IoT. Solutions are grouped based on the mentioned classification in Fig. 1.

					Confidentiality	Integrity	Authentication	Authorization	Freshness	Resilience	Computation Complexity	Communication Complexity	Memory	Scalability	Privacy Protection
Key bootstrapping in IoT	Asymmetric key schemes	Key transport based on public key encryption	RPKE	Protocols based on: NTRU [44], Rabin's scheme	●	n/a	n/a	n/a	n/a	●	○	○	●	●	n/a
				Moustaine and Laurent [56]	n/a	●	●	○	○	●	●	●	●	●	●
				ZKP based on ECDLP [34]	n/a	●	●	○	●	●	○	●	●	●	●
			CBE	DTLS modified [4,36]	●	●	●	○	●	●	○	○	●	●	●
				IKEv2-ECC based [59]	●	●	●	○	●	●	○	○	●	●	●
			IBS	TinyIBE [7]	●	○	●	○	○	●	●	○	●	○	○
				IBAKA [46]	●	●	●	○	●	●	○	○	●	●	●
		Key agreement based on asymmetric techniques	ECDH-ECDSA [32]		●	●	●	○	●	●	○	●	●	●	○
			HIP-DEX [15]		●	●	●	○	●	●	●	●	●	●	○
	Symmetric key pre-distribution schemes	Probabilistic key distribution	E-G [48]		●	○	○	○	○	○	●	○	○	○	○
			Du et al. [37]		●	○	○	○	●	○	●	○	○	○	○
			Chan et al. [49]		●	○	○	○	○	○	●	○	○	○	○
			Ito et al. [50]		●	○	○	○	○	○	●	○	○	○	○
		Deterministic key distribution	OKD	Blom's scheme based [37,38]	●	○	○	○	○	○	●	●	●	○	○
				SNAKE [41]	●	●	●	○	●	○	●	●	●	○	○
				BROSK [40]	●	●	●	○	●	○	●	●	●	○	○
				Lightweight IPsec [25]	●	●	●	○	●	○	○	○	●	●	●
				DTLS-PSK [31]	●	●	●	○	●	●	○	○	●	●	●
				Diet-ESP [55]	●	●	●	○	●	●	●	○	n/a	○	●
			EAS	Mikey-ticket [17]	●	●	●	●	●	○	●	●	●	○	○
				SAKE [20]	●	●	●	○	●	●	●	●	●	○	○
				PANA/EAP-PSK [23,24]	●	●	●	○	●	●	●	●	●	○	○
			PBAS	SAKES [42]	●	●	●	○	●	●	○	●	○	○	○
				D-HIP [43]	●	●	●	○	●	○	○	○	○	○	○

Some abbreviations are used: (RPKE) – Raw public key encryption, (CBE) – Certificate based encryption, (IBS) – Identity based schemes, (OKD) – Offline key distribution, (EAS) – External server assisted, (PBAS) – Proxy-based assisted server. Eleven metrics are provided to evaluate the solutions: Confidentiality, Integrity, Authentication, Authorization, Freshness, Resilience, Computation complexity, Communication Complexity, Memory or storage space required for keying materials, Scalability and Privacy Protection. The Resilience, Computation complexity, Communication Complexity and Memory columns can take two different values: ● (good or medium performance level) and ○ (low performance level), which indicate the level of a specific protocol to support a property. Communication complexity refers to the number of message exchanges in general until a secret key is negotiated. The (n/a) notation means "not applicable". We define simple notations to evaluate the security services: ● – *supported,* ○ – *not supported.* The evaluation of RPKE assumes the protocols that used the mentioned primitives (no real protocol reference).

high memory space for keying materials, low level of scalability for large networks and vulnerability against node capture attacks.

7. Overview on recent trends on IoT security protocols

There are some new approaches being pushed by researchers. They always keep their interest in both asymmetric and symmetric approaches; even if the symmetric paradigm is considered to be more energy efficient. The asymmetric solutions are still preferable because of their deployment facility, flexibility and scalability in terms of key management. Besides, the public key paradigm allows two entities without any prior-trust relationship with each other, establishing a secure channel, which is generally an important feature in real time scenarios.

The following points need to be highlighted before designing any efficient security protocols for constrained devices in IoT:

Optimizing asymmetric solutions: The asymmetric approaches are generally energy-consuming. The first ambition is to reduce the required computation time in order to save energy for sensor nodes. One can think about adapting directly NTRU to the standard protocols because it is currently the most energy-efficient primitive. However, this primitive requires more memory space for keying materials than other asymmetric primitives. Some researchers are working on optimizing mathematical mechanisms used in cryptographic algorithms, i.e. Marin et al. [35] discuss a solution to optimize the ECC primitives. They propose an optimization for the modular multiplication operation. The solution is evaluated in the widely-used microprocessor MSP430. The authors claimed that the optimization is presenting the lowest time and number of required operations for ECC multiplication. Another method to reduce the energy consumption on sensor nodes relies on pre-computation techniques. It helps diminishing the cost of modular exponentiations in several signature and key management schemes, such as ECDSA or Diffie–Hellman key exchange. The idea is to store a set of n Discrete Log pairs in the form $(a_i, g^{a_i} \ mod \ q)$. Then, a "random" pair $(r, g^r \ mod \ q)$ is generated from a subset of k pairs chosen randomly in the memory. The technique seems simple, but it requires the value of n to be sufficiently large in order to ensure the randomness of the generated pairs $(r, g^r \ mod \ q)$. Ateniese et al. [65] improve the pre-computation techniques above and apply it to ECDSA. They show that the almost 50% of energy is saved with ECDSA with pre-computation compared to the original signature scheme and also to the *NTRUsign* signature scheme (which is considered to be a natural candidate in low-power devices).

On the other hand, several researches adapt the properties of asymmetric primitives in an optimized manner to fit in the most constrained environment of IoT. Effectively, Moustaine and Laurent [56] propose an efficient authentication protocol for low-cost RFID systems based on an adaption of NTRU. This adaption first delegates the complex operations of NTRU (i.e. modular arithmetic, polynomial multiplication) to the server. Secondly, the tags require only additions and circular shifts to encrypt the challenges during the authentication phase. Besides, the protocol is resistant against classical attacks including replays, tracking and man in the middle attacks with very low requirements for computation.

As another asymmetric technique, Zero-knowledge proofs (ZKP) [16,34] is also a candidate for future proposals in IoT. ZKP are interactive proof systems involving two entities: a prover and a verifier. The prover demonstrates the knowledge of a secret to the verifier without revealing a single bit about the secret. ZKP relies on some hard mathematical problems, such as the factorization of integers, i.e. [16] or the discrete logarithm problem (DLP) [34]. This mechanism is commonly used in WSN for node authentication. For example, the authors in [34] provide an efficient authentication scheme based on DLP over elliptic curve groups. The scheme requires only three messages between the prover and verifier. ZKP has advantages in terms of the amount of messages being sent and the memory usage on nodes as also mentioned in [16,34]. One can benefit ZKP to propose an efficient key bootstrapping protocol in IoT with the node authentication provided by ZKP.

Tailoring the existing standard protocols to IoT: Standard security protocols can be adapted to work in constrained and heterogeneous environments of IoT. As described in this document, many attempts have been done to adapt and apply standard protocols in the context of IoT, for example, DTLS [4,36], IPsec [25], IKEv2 [59], HIP-DEX [2,15,53]. As another example, Kivinen [57] propose a minimum implementation of standard IKE [58] by removing the requirement for certificates. This minimum variant defines only two message exchanges for key negotiation and provides entity authentications using pre-shared key approach. On the other hand, Migault et al. [55] suppose that the security associations between entities are established using existing mechanism like IKEv2. They are interested in the security of packet transmissions by proposing Diet-ESP – an adaptation of ESP (Encapsulation Security Protocol) to IoT in order to compress and reduce the ESP overhead. The authors define mechanisms to remove or reduce some "*unnecessary*" or "*larger than required*" ESP fields for the specific needs or applications of IoT devices. However, the deployment of Diet-ESP has to keep the trade-off between the security requirements and the battery life time of constrained devices. Indeed, as depicted by the authors, small SPI (Security Parameters Index) size, small size of ICV (Integrity Check Value) and removing SN (Sequence Number) expose the devices to respectively Denial of Service, spoofing and replay attacks.

Using hybrid approaches: Another trend consists of combining the advantages of both symmetric and asymmetric solutions. Meca et al. [2] choose HIP-DEX (an asymmetric technique) [15] to provide access to a local sensor network. A mobile node is authenticated with help of a central server. If the authentication is successful, the server sends securely the necessary parameters for the mobile node by encrypting the data with the session key generated after the DH exchanges. These parameters are actually a bivariate polynomial used to bootstrap secure communications with a local node (a symmetric technique). The pairwise key generated by the shared polynomial is employed

as a master key to generate multiple session keys for specific purposes.

The presence of a third party in such hybrid approach becomes essential in the IoT. Firstly, the rich-resource server is expected to support almost all heavyweight computations. As such, the sensor nodes with limited energy and capabilities are no longer involved in this expensive process as described in [42,43]. The constrained node can establish a communication with external hosts without implementing the full asymmetric process. Additionally, the assisted servers are capable to provide fine-grained access control such that only authorized actions are executed on sensor nodes.

8. Conclusion

This paper studied multiple secure, lightweight and attack-resistant solutions for WSNs and IoT based on identified security requirements and challenges. We also provided a novel classification of existing protocols relying on their key bootstrapping approach to establish a secure communication channel. These protocols and techniques are analyzed according to different criteria in order to identify the advantages and drawbacks of each protocol.

Using this methodology, we noted that symmetric approaches are not anymore the default choice for IoT. Public key cryptography is likely to be increasingly recommended in the IoT context, provided that the associated asymmetric techniques are properly optimized. A trusted third party will also certainly take a more active role to secure the IoT and to adapt to its heterogeneous nature. Additionally, security protocols should take into account the resource-constrained feature of *things*. Heavyweight cryptographic operations i.e. based on RSA and Diffie–Hellman agreement protocols should be replaced by lightweight operations, i.e. using symmetric cryptography or applying more lightweight asymmetric primitives such as ECC and NTRU. Besides, lightweight security protocols are also needed to reduce the communication complexity. Aside from performance concerns, the future proposed security solutions will offer perspectives on new applications that increasingly expand the coverage of capabilities and features offered by IoT devices making them more and more intelligent.

References

[1] O. Garcia-Morchon, S. Kumar, Security Consideration in the IP-based Internet of Things, CoRE, Internet-draft, 2013.

[2] F. Meca, J. Ziegeldorf, et al., HIP security architecture for the IP-based Internet of thing, in: 27th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2013.

[3] T. Kothmayr, C. Schimit, et al., A DTLS based end-to-end security architecture for the Internet of thing with two-way authentication, in: 7th IEEE International Workshop on Practical Issues in Building Sensor Network Applications, 2012.

[4] S. Raza, H. Shafagh, et al., Lithe: lightweight secure CoAPs for the Internet of things, IEEE Sens. J. 13 (10) (2013).

[5] B. Aoba, L. Blunk, et al., Extensible Authentication Protocol (EAP), IETF, RFC 3748, June 2004.

[6] A. Shamir, Identity-based cryptosystems and signature schemes, in: Proc. Crypto'84, Santa Barbara, California, USA, 1984, pp. 47–54.

[7] P. Szczechowiak, M. Collier, TinyIBE: identity-based encryption for heterogeneous sensor networks, in: 5th International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), 2009.

[8] A. Fanian, M. Berenjkoub, H. Saidi, A scalable and efficient key establishment protocol for wireless sensor networks, in: IEEE Globecom Workshop on Web and Pervasive Security, 2010.

[9] HP report on Internet of Things Research Study, 2014. <http://fortifyprotect.com/HP_IoT_Research_Study.pdf>.

[10] Thesis: Collaborative Security for the Internet of Thing, Yosra Ben Saied. <http://www.theses.fr/2013TELE0013> (accessed November 2013).

[11] E. Rescorla, Diffie–Hellman Key Agreement Method, IETF, RFC 2631, 1999.

[12] S. Turner, T. Polk, Transport Layer Security, IETF, RFC 6176, 2011.

[13] P. Eronen, H. Tschofenig, Pre-Shared Key Ciphersuites for Transport Layer Security (TLS), IETF, RFC 4279, 2005.

[14] J. Hui, P. Thubert, Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, IETF, RFC 6282, 2011.

[15] R. Moskowitz, HIP Diet EXchange (DEX), IETF, draft-moskowitz-hip-rg-dex-06, 2012.

[16] I. Chatzigiannakis, A. Pyrgelis, et al., Elliptic curve based zero knowledge proofs and their applicability on resource constrained devices, in: 8th IEEE Conference on Mobile Ad-Hoc and Sensor Systems, 2011.

[17] J. Mattsson, T. Tian, MIKEY-TICKET: Ticket-Based Modes of Key Distribution in Multimedia Internet KEYing (MIKEY), IETF, RFC 6043, March 2011.

[18] J. Kohl, C. Neuman, The Kerberos Network Authentication Service (V5), IETF, RFC 4120, 6649, July 2005.

[19] M.O. Rabin, Digitalized Signature and Public Key Functions as Intractable as Factorization, MIT/LCS/TR-212, Massachusetts Institute of Technology, 1979.

[20] A. Boudguiga, A. Olivereau, N. Oualha, Server assisted key establishment protocol for WSN: a MIKEY-ticket approach, in: 12th IEEE Trustcom, 2013.

[21] Gartner Inc., Forecast: The Internet of Things, Worldwide, 2013.

[22] J. Arkko, E. Carrara, F. Lindholm, et al., MIKEY: Multimedia Internet KEYing, RFC 3830, August 2004.

[23] D. Forsberg, Y. Ohba, et al. (Eds.), Protocol for Carrying Authentication for Network Access (PANA), RFC 5191, 2008.

[24] M. Kanda, Y. Ohba, S. Das, et al., PANA Applicability in Constrained Environments, Sources. <http://www.lix.polytechnique.fr/> (accessed February 2012).

[25] S. Raza, S. Duquennoy, T. Chung, et al., Securing communication in 6LoWPAN with compressed IPsec, in: International Conference on Distributed Computing in Sensor Systems and Workshop, 2011.

[26] D. Liu, P. Ning, R. Li, Establishing pairwise keys in distributed sensor networks, J. ACM Trans. Inform. Syst. Secur. 8 (1) (2005) 41–77.

[27] G. Gaubatz, J.-P. Kaps, B. Sunar, State of the art in ultra-low power public key cryptography for wireless sensor networks in: 3rd IEEE International Conference on Pervasive Computing and Communications Workshop (PERCOMW), 2005.

[28] Proofpoint, Article Proofpoint Uncovers Internet of Things (IoT) Cyberattack. <http://www.proofpoint.com/about-us/press-releases/01162014.php> (accessed September 2014).

[29] C. Bormann, M. Ersue, A. Keranen, Terminology for Constrained Node Networks, Draft-Internet, 2013.

[30] NSA, The Case for Elliptic Curve Cryptography. <http://www.nsa.gov/business/programs/elliptic_curve.shtml> (accessed February 2014).

[31] J. Granjal, E. Monteiro, J. Silva, End-to-end transport layer security for Internet-integrated sensing applications with ECC public-key authentication, in: IFIP Networking Conference, 2013.

[32] G. de Meulenaer, F. Gosset, et al., On the energy cost of communication and cryptography in wireless sensor network, in: IEEE International Conference on Wireless and Mobile Computing, Network & Communication, 2008.

[33] W. River, White Paper, Security in the Internet of Things, 2014. <http://www.windriver.com/whitepapers/security-in-the-internet-of-things/wr_security-in-the-internet-of-things.pdf>.

[34] U. Feige, A. Fiat, A. Shamir, Zero-knowledge proofs of identity, J. Cryptogr. 1 (2) (1988).

[35] L. Marin, A. Jara, et al., Shifting primes: optimizing elliptic curve cryptography for smart things, in: 6th International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2012.

[36] R. Hummen, Jan H. Ziegeldorf, et al., 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), 2013.

[37] W. Du, J. Deng, et al., A key predistribution scheme for sensor networks using deployment knowledge, IEEE Trans. Dependable Secure Comput. 3 (1) (2006) 41–77.
[38] R. Blom, An optimal class of symmetric key generation systems, in: Advances in Cryptology: Proc. EUROCRYPT '84, 1985, pp. 335–338.
[39] A. Perrig, R. Szewczyk, J.D Tygar, et al., SPINS: security protocols for sensor networks, in: ACM International Conference on Mobile Computing and Networking (MobiCom), 2001.
[40] B. Lai, S. Kim and I. Verbauwhede, Scalable session key construction protocol for wireless sensor networks, in: IEEE Workshop on Large Scale RealTime and Embedded Systems (LARTES), 2002.
[41] S. Seys, Key Establishment and Authentication Suite to Counter DoS Attacks in Distributed Sensor Networks, COSIC, 2012.
[42] H.R. Hussen, G.A. Tizazu, et al., SAKES: secure authentication and key establishment scheme for M2M communication in the IP-based wireless sensor network (6LoWPAN), in: 5th International Conference on Ubiquitous and Future Networks (ICUFN), 2013.
[43] Y.B. Saied, A. Olivereau, D-HIP: a distributed key exchange scheme for HIP-based Internet of things, in: First IEEE WoWMoM Workshop on the Internet of Things: Smart Objects and Services (IoT-SoS), 2012.
[44] B. Sarikaya, Y. Ohba, et al., Security Bootstrapping Solutions for Resource-Constrained Devices, Internet-draft, 2012.
[45] D. Boneh, M. Franklin, Identity-based encryption from the Weil paring, SIAM J. Comput. 32 (2003) 586–615.
[46] L. Yang, C. Ding, M. Wu, Establishing Authenticated Pairwise Key using Pairing-based Cryptography for Sensor Network, 8th Chinacom, 2013.
[47] C. Gentry, Practical identity-based encryption without random oracles, in: Proc. of the EUROCRYPTO'06, Springer-Verlag, 2006, pp. 445–464.
[48] L. Eschenauer, V.D. Gligor, A Key-Management Scheme for Distributed Sensor Networks, 2002.
[49] H. Chan, A. Perrig, D. Song, Random key predistribution schemes for sensor networks, in: Proc. IEEE Symp. Security and Privacy, May 2003.
[50] T. Ito, H. Ohta, et al., A key pre-distribution scheme for secure sensor networks using probability density function of node deployment, in: Proc. 3rd ACM Workshop on Security and Ad Hoc Sensor Networks, 2005, pp. 69–75.
[51] D.D. Hwang, B. Lai, I. Verbauwhede, Energy-memory-security tradeoff distributed sensor networks, in: Proc. 3rd Conference on Ad-Hoc Networks and Wireless, 2004, pp. 70–81.
[52] S. A. Camtepe, B. Yener, Key Distribution Mechanisms for Wireless Sensor Networks: A Survey, Technical Report TR-05-07, Rensselaer Polytechnic Institute, 2005.
[53] R. Moskowitz, P. Jokela, et al., Host Identity Protocol version 2 (HIPv2), Draft-Internet, 2013.
[54] R. Hummen, H. Wirtz, et al., Tailoring end-to-end IP security protocols to the Internet of things, in: 21th International Conference on Network Protocols (ICNP), 2013.
[55] D. Migault, T. Guggemos, D. Palomares, Diet-ESP: A Flexible and Compressed Format for IPsec/ESP, Draft-Internet, January 2014.
[56] E. Moustaine, M. Laurent, A lattice based authentication for low-cost RFID, in: IEEE International Conference on RFID-Technologies and Applications (RFID-TA), 2012.
[57] T. Kivinen, Minimal IKEv2, Draft-Internet, 2011.
[58] C. Kaufman, Internet Key Exchange (IKEv2) Protocol, IETF, RFC 4306, 2005.
[59] S. Ray, G.P. Biswas, Establishment of ECC-based initial secrecy usable for IKE implementation, in: Proc. of World Congress on Expert Systems (WCE), 2012.
[60] D. Miorandi, S. Sicari, F.D. Pellegrini, Internet of things: vision, applications and research challenges, Ad Hoc Netw. 10 (7) (2012) 1497–1516.
[61] L. Atzori, A. Iera, G. Morabito, The Internet of things: a survey, Comput. Netw. 54 (15) (2010) 2787–2805.
[62] J.S. Kumar, D.R. Patel, A survey on Internet of things: security and privacy issues, Int. J. Comput. Appl. 90 (11) (2014) 20–26.
[63] R. Roman, C. Alcaraz, et al., Key management systems for sensor networks in the context of the Internet of things, Int. J. Comput. Electr. Eng. (2011) 147–159.
[64] Y. Wang, G. Attebury, B. Ramamurthy, A survey of security issues in wireless sensor networks, IEEE Commun. Surv. Tutorials 8 (2) (2006).
[65] G. Ateniese, G. Bianchi, et al., Low-cost Standard Signature in Wireless Sensor Networks: A Case for Reviving Pre-computation Techniques, Usenix Network and Distributed System Security Symposium (NDSS), 2013.

Kim Thuat Nguyen is currently pursuing his PHD thesis at the Communicating Systems Laboratory at CEA, France. He has obtained his engineering diploma from the engineering school of Informatics, Applied Mathematics and Telecommunications (Ensimag), Grenoble INP, France. His research focus on lightweight security protocols for IP-based Wireless Sensor Networks and the Internet of Things.

Maryline Laurent, PhD works as a professor at Telecom SudParis, Mines-Telecom Institute, and is the head of the research team R3S (Network, Systems, Services, Security) of the French CNRS UMR 5157 SAMOVAR. Her main topics of interest are related to network security and personal data privacy, including clouds, tiny devices (RFID, sensors), social networks and digital identity management. She cofounded the chair of Institut Mines-Télécom on Values and personal information policies. She chaired several workshops and conferences like Data Privacy Management DPM 2013, Secure Smart Objects SSO 2013, and IFIP New Technologies, Mobility and Security NTMS 2011. She is co-editor of the special issue on « Privacy-aware electronic society », Annals of Telecommunication, 2014. She published a number of papers in journals, international conferences and book chapters.

Nouha Oualha is a research engineer at the Communication Systems Laboratory of CEA-LIST, since October 2010. She graduated from the engineering school of telecommunications, Telecom Bretagne (formerly known as ENST Bretagne), in 2005. She received her PhD degree from Telecom ParisTech (formerly known as ENST) in 2009, on the topic of "Security and cooperation for peer-to-peer data storage". Her research interests focus on several topics such as secure systems, security protocols and cryptographic algorithms, as well as, peer-to-peer and IP-based wireless sensor networks.