IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 22, 2019, accepted February 10, 2019, date of publication February 21, 2019, date of current version March 13, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2900957

A Survey of Key Bootstrapping Protocols Based on Public Key Cryptography in the Internet of Things

MANISHA MALIK[1], MAITREYEE DUTTA[1], AND JORGE GRANJAL[2], (Member, IEEE)
[1]National Institute of Technical Teachers Training and Research, Chandigarh 600116, India
[2]Centre for Informatics and Systems, University of Coimbra, 3030-290 Coimbra, Portugal

Corresponding author: Jorge Granjal (jgranjal@dei.uc.pt)

This work was supported in part by the MobiWise project (P2020 SAICTPAC/0011/2015), in part by the COMPETE 2020, in part by the Portugal 2020 - Operational Program for Competitiveness and Internationalization (POCI), in part by the European Union European Regional Development Fund (ERDF), and in part by the Portuguese Foundation for Science and Technology (FCT).

ABSTRACT The Internet of Things envisages connecting all physical objects or things to the Internet, using devices as diverse as smartphones, coffee makers, washing machines, automobiles, lamps, and wearable devices, among many others. The explosive growth of Internet-connected sensing and actuating devices has bridged the gap between the physical and the digital world, with new solutions bringing benefits to people, processes, and businesses. However, security will be a major challenge in enabling most of such applications. The lack of secure links exposes data exchanged by devices to theft and attacks, with hackers already showing a keen interest in this area. Secure communication in the IoT will require a multifaceted approach, in particular, targeting aspects as relevant as the communications' protocols and data that need to be secured. One of the major aspects among these is how keys are bootstrapped in devices, for the purpose of supporting secure communications. In this paper, we survey the state of the art in key bootstrapping protocols based on public-key cryptography in the Internet of Things. Due to its inherent scalability, such protocols are particularly relevant for the implementation of distributed identity and trust management mechanisms on the IoT, in the context of which devices may be authenticated and trusted. The reviewed proposals are analyzed and classified on the basis of the key delivery method, the underlying cryptographic primitive, and the authentication mechanism supported. We also identify and discuss the main challenges of implementing such methods in the context of IoT applications and devices, together with the main avenues for conducting further research in the area.

INDEX TERMS Authentication, Internet of Things, key bootstrapping, key management, public key cryptography, security.

I. INTRODUCTION

The Internet of things (IoT) is touted to be one of the key enablers of the next revolution in the digital world. The IoT allows to connect everyday objects (or things) to the Internet, by equipping such devices with identifying, sensing, networking and processing capabilities. Such capabilities allows objects with sensing and actuating capabilities to communicate with each other, and also with other devices and services over the Internet, in order to accomplish tasks in the context of IoT applications. Areas for new IoT applications include smart homes, intelligent transportation systems, smart buildings and smart environment monitoring system, among others. The IoT is poised for explosive growth, with about 50 billion smart devices connected to the Internet by 2020, and estimated to generate over $1.7 trillion revenue per year [1]. Although the seeds of the term IoT were planted by British entrepreneur Kevin Ashton in the year 1999, while working at the MIT Auto-ID Center, its recent rise is being fueled by the advancement of digital technologies, such as low-cost while highly capable sensors and processors, efficient wireless protocols, the mobile revolution and a myriad of startups and established companies developing the necessary application and management software.

The IoT is the convergence of the cyber and physical worlds, with the goal of creating an open and global network

The associate editor coordinating the review of this manuscript and approving it for publication was Congduan Li.

to connect people, things, and data. Simply put, an IoT network is made up of a great number of heterogeneous devices and technologies, produced by different vendors and for different purposes, also characterized by different capabilities. These devices have multi-faceted constraints in terms of processing capability, memory, power supply, communication capability and user interfaces [2]. The use of constrained devices in networks often also leads to constraints on the networks themselves. However, there may also be constraints on networks that are largely independent of those of the nodes. These constraints include high packet loss, low achievable throughput, lack of advanced security services and highly asymmetric links, among others. The main challenge is to adapt such networks to operate in the conventional Internet, and the integration of Wireless Sensor Networks (WSN) with the Internet communications infrastructure is a required and strategic step in the right direction. Hence, in this context, research and development challenges are enormous, and this certainly applies to security. Some of these challenges are massive scaling, openness, robustness, architecture and its dependencies, big data, security and privacy [3]. Other challenges, such as the lack of standards, interoperability, legal issues and the cultural impact related with the usage of such technologies also inhibit the realization of IoT applications.

Security for IoT is still in its infancy, and to ensure the security of communications, recent research has addressed the usage of adaptation of standard protocol solutions such as IPSec/IKE, MIKEY, TLS, DTLS and HIP to IoT environments [4]. However, the expensive and resource consuming operations of such solutions, the constrained nature and the scalability of IoT devices, hamper its direct implementation in most IoT applications. Among all security challenges that are certainly arising in the IoT, one of the challenges that has been daunting researchers and industry personnel is how to bootstrap security associations among nodes in IoT. As we observe throughout our discussion in this article, many security protocols are being designed to secure communications in the IoT, but without specifying how the required cryptographic keys are configured in the intervening devices, in the first place. The security association includes attributes like cryptographic algorithm and its mode, the cryptographic key and other network parameters, required to establish a secure connection. The management of these cryptographic keys in a cryptosystem belongs in the context of key management, which is one of the most difficult aspects of cyber security. Key management includes the generation, exchange, storage, usage and replacement of keys. Our focus in the survey is on solutions to support key generation and exchange, in the context of IoT applications. Key bootstrapping thus involves generation and exchange of cryptographic keys, and is a vital component of an overall IoT key management solution. After the bootstrapping phase of the key management process, the keys thus established may be employed to satisfy security services such as encryption, authentication, non-repudiation and digital signatures.

In the recent years, a lot of research focused on the challenge of securely assuring key bootstrapping in constrained IoT environments. While key bootstrapping solutions based on symmetric cryptography (as with pre-shared secret models, for example employed in Kerberos) are simple and incur on less overhead, public key cryptography based solutions (based on protocols such as RSA or ECC [5]) are more scalable, and as such more in line with the need to assure key distribution to millions of devices. Public Key Cryptography (PKC) based key bootstrapping solutions are also more robust to attacks than their symmetric counterparts. The main motivations of this survey are to provide a taxonomy of public key cryptography based key bootstrapping protocols for the IoT, while also identifying open research challenges that provide open avenues for research. The reviewed literature is classified, analyzed and compared according to different evaluation metrics. During our analysis, a number of issues were discovered in the proposed schemes and thus, we also present possible research areas that address those issues.

A. PAPER OUTLINE

The outline of the article is as follows. In Section II we discuss security in the IoT, particularly the significance of key bootstrapping and of asymmetric cryptography in the context of the life cycle of an IoT sensing device. Section III discusses the enabling technologies and protocols for IoT, in the context of which we discuss key bootstrapping approaches and proposals throughout the article. Section IV presents the proposed taxonomy and discusses relevant work regarding the classification of key bootstrapping solutions for the IoT. Section V examines in depth the existing proposals in the context of the considered taxonomy, which are further analyzed and discussed in Section VI. Finally, Section VII identifies recent trends and opportunities for further research in the area, and Section VIII concludes the article.

II. SECURITY IN THE IoT

Key bootstrapping, and in particular using asymmetric cryptography, plays an important role in the context of IoT security. It is also relevant to explore the importance of key management and bootstrapping considering the lifecycle of an IoT device, as we proceed to discuss.

A. THE LIFECYCLE OF AN IoT DEVICE

The lifecycle of a device (or thing) in IoT is composed of the bootstrapping, operational and maintenance phases [6]. It is also important to note that, due to the varied application areas, it is highly unlikely that all nodes will be manufactured by a single manufacturer. This raises challenges on how to approach the bootstrapping of the required security material in such devices, in order to ensure interoperability and trusted communication between nodes, from the start of operations. The execution of an application corresponds to the operational phase, in which the device is under the control of the system owner, but we need to consider also other phases. In the maintenance phase, the device's software

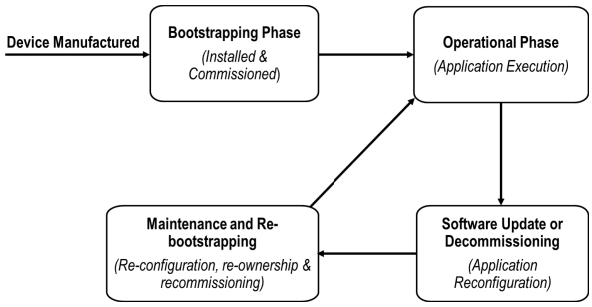

FIGURE 1. The lifecycle of a thing in IoT.

can be upgraded, or applications running on the device can be reconfigured. This way the device continues to loop through operational and maintenance phases, until it is decommissioned at the end of its lifecycle, as Figure 1 illustrates.

We need to consider that security, given that it is a key requirement of the IoT, must be ensured throughout all the phases of this life cycle. The first and most important phase is bootstrapping since, as discussed previously, most existing IoT security proposals depend on the availability, from the start, of the keying material required to secure communications, while not defining specific solutions on how this may be guaranteed. In general, the role of bootstrapping refers to any processing required before a network, device or system is able to start operating in the context of a given IoT application. Typically, it involves establishing an association between devices that have no apriori information about each other. Bootstrapping also serves the purpose of establishing trust between such devices.

It is also important to note that, in the context of key bootstrapping, the usage of Public Key Protocols is of particular interest, since this allows security to really scale to the huge number of devices which are expected to be employed in the IoT. The usage of public key cryptography is particularly challenging in constrained sensing environments, but key bootstrapping technologies based on asymmetric cryptography will be required in the IoT. As is rightly stated in Kerchoff's principle [7], the security of any cryptosystem should lie solely in its keys with everything else, including the cryptographic algorithm, considered to be public knowledge. Although key management includes other principles, in this work we focus on the secure generation and exchange of keys to setup security in the context of IoT applications.

B. KEY BOOTSTRAPPING IN THE CONTEXT OF THE IoT

Security bootstrapping refers, in general, to creating a security association between two or more devices in a network. However, the term has often been used in a number of different contexts. For instance, [6] defines it as a process by which a thing securely joins an IoT system at a given location and point in time. He and Sarikaya [8] define it as a way to authenticate the identity of devices and to transfer security credentials and other keying materials in order to establish trust relationships between devices. Other functions include authorization for network access, registration to join a group, or pairing with a specific node. Even in [9], key bootstrapping is defined as a process in which a device is associated with another device, system or a network. Similarly, Sarikaya *et al.* [9] define it as a prerequisite before any network can operate, and which involves the configuration of various settings at the application layer (network names, application encryption keys) or at the link layer (wireless channels, link-layer encryption keys).

In this article we consider secure key bootstrapping in the context of IoT environments, where the term bootstrapping is used to refer to the generation and exchange of keying materials between unassociated devices. Key bootstrapping is certainly a critical phase in the security management of an IoT application, and is the focus of our discussion in the article. As we discuss later, most of the existing proposals on security protocols and mechanisms for the IoT do not address this goal, and this motivates us in identifying and surveying existing and future research approaches in this area. In particular, we focus on proposals towards the introduction of such mechanisms based on the usage of asymmetric cryptography, due to their appropriateness to support upcoming IoT applications.

C. ON THE USAGE OF ASYMMETRIC CRYPTOGRAPHY ON THE IoT

Public key cryptosystems are based on expensive asymmetric cryptographic operations, and due to this fact they are not usually employed to directly encrypt large blocks of data by constrained sensing devices. On the other hand, such solutions may be used to encrypt smaller blocks of data, in particular those employed to transport secret keys, in the context of a key distribution protocol. Such systems may be based on algorithms such as RSA and ECC, or employ Diffie-Hellman (DH) [10] for the purpose of key negotiation and exchange. The security of the RSA algorithms relies on the hard mathematical problems of prime factorization, while ECC is based on the elliptic curve discrete logarithm problem. On the other hand, DH security rests on the discrete logarithm problem. Traditional Diffie-Hellman key exchange (DHKE) solutions suffers from the Man-in-the-Middle (MITM) attack, and as such secure variants such as the STS [11] protocol may be used to secure communication between devices. Discovered in 1985 by mathematicians Neil Koblitz and Victor Miller, the shorter key length of ECC enables it to meet the security requirements of virtually any application.

Of all known asymmetric key algorithms, ECC provides the highest strength per key bit, thus it can offer security similar to RSA, while with a smaller key, an important advantage for the enabling of practical IoT applications with sensing and actuating devices. A general performance comparison of RSA and ECC digital signatures is provided in [12]. Later, Kothmayr *et al.* [13] ported the ECC and RSA implementation of CyaSSL project to TinyOS [14] and evaluated its performance. The authors concluded that for a DTLS handshake,

the computation time and energy consumption of RSA was much greater than that of ECC.

To achieve end-to-end security specifically between users of real-time and multimedia applications, the MIKEY protocol was initially proposed in [15] and for IoT in [16]. MIKEY enables sharing of session keys and authentication of users. It supports various modes which are loosely based on the concepts of pre-shared keys, public-keys, diffie-hellman key exchange, identities and tickets. MIKEY was designed to have characteristics similar to those of the constrained sensing devices enabling IoT applications, such as low computation, low bandwidth, smaller code size and minimal round-trips. As MIKEY was deployed, a number of extensions to the conventional protocol emerged. The extensions based on ECC have been particularly popular because ECC supports smaller key sizes than RSA, and is thus more suitable for constrained devices. As already discussed, ECC offers encryption, authentication and digital signatures to achieve secure key distribution. Another variant is the Host Identity Protocol (HIP) [17], a new protocol layer between the transport and network layers. It is an identification technology which has been used in many authentication systems. The cryptographic exchange in the HIP architecture is also referred to as the HIP Base exchange.

Although asymmetric key schemes have low memory requirements, high scalability and resilience to attacks, they employ computationally intensive operations which increase the energy consumption and computation cost in the context of IoT applications. However, in the recent years, a lot of research work has been focused on optimizing the expensive operations of PKC for IoT devices. In our discussion throughout the article, we focus on such proposals, while considering the taxonomy presented later in the survey to guide our analysis.

III. SECURITY ANALYSIS OF THE PROTOCOLS OF IoT

The vision of the IoT brings together diverse communication technologies, and this encompass Wireless Sensor Networks (WSN), RFID, NFC, Bluetooth, GSM, Wi-Fi and Internet Protocol version 6 (IPv6), among others. In reality, WSN and IPv6 in particular are proving to be key enabling technologies for the IoT, particularly as research and standardization efforts are materializing the integration of WSN devices with the Internet infrastructure [4]. WSN environments, which were once designed to support isolated sensing and actuation applications, are now starting to be integrated with the Internet communications infrastructure, an effort progressing towards the support of IoT applications, also fueled by the upcoming 5G architecture [18]. From the start of this process, research and standardization efforts from bodies such as the Institute of Electrical and Electronics Engineers (IEEE) and the Internet Engineering Task Force (IETF) were of prime importance. One major milestone in this context was the design of the 6LoWPAN (IPv6 over Low-Power Wireless Personal Area Networks) adaptation layer by the IETF [19], as it lays the ground for the usage of standard

IoT Layer	IoT Protocol	Security Protocol	
Application Layer	CoAP	DTLS	Cross-Layer Security Mechanisms (Key management, fault tolerance)
Transport Layer	UDP		
Network and routing layer	IPv6, RPL	IPSec, RPL security	
Adaptation Layer	6LoWPAN	None	
Data link layer	IEEE 802.15.4	IEEE 802.15.4 security	
Physical layer			

FIGURE 2. A standardized communications and security stack for the IoT.

IPv6 communications on constrained communication environments using heterogeneous communication technologies, in fact forming a standardized protocol stack [20]. In Figure 2 we illustrate the protocol stack developed by the IETF and most commonly used by academia to support research efforts in the IoT. This figure also identifies the security approaches considered in the context of such technologies.

It is important to note that, in our analysis throughout the article, we focus on the low-energy and short-range communication technologies being developed in the context of the previously presented stack, thus to support end-to-end communications between constrained sensing devices and other entities in the Internet infrastructure, using the IP communications protocol. As we can observe in Figure 2, the security mechanisms adopted target the protection of communications at a particular layer of the stack. Key bootstrapping (and key management for that mater) is a cross-layer requirement, and one that is currently not properly addressed in the context of the communication and security protocols. This is due to the fact that security protocols such as DTLS and IPSec (as adapted to 6LoWPAN environments [4]) define modes for the application of security to its communications, but leave absent how the keys required for the application of security are established in the first place. As we discuss throughout the article, key bootstrapping is a fundamental aspect to guarantee the secure operation of the network, since it is necessary to equip the IoT devices with the cryptographic material required to support security communications and operations during the lifetime of applications. We proceed by discussing in greater detail the role of security, and in particular of key bootstrapping identified as cross layer security challenge, in the communication and security protocols as shown in Figure 2.

A. DATA LINK AND PHYSICAL LAYER PROTOCOLS

At the physical and link layers, the IEEE 802.15.4 [21] standard provides the support for the usage of Zigbee [22] and WirelessHART [23], as well as 6LoWPAN, each of which extends the standard by developing the upper layers. The IEEE 802.15.4 link layer standard provides security but only for hop-by-hop communication, since it does not address end-to-end security, as will be required for IoT applications. The standard only defines mechanisms to provide hop-by-hop security, while considering that the required cryptographic keys are already available in the memory of the device. The bootstrapping of the required keying material, together with the absence of a specific keying model, are major challenges in what respects the enabling of security at the IEEE

802.15.4 layer. Other limitations and research issues of IEEE 802.15.4 security are the management of the initialization vector (IV) values, the implementation of group and network shared key management and the fact that IEEE 802.15.4 does not secure acknowledgment messages [4].

B. 6LOWPAN ADAPTATION LAYER

Following the path to realizing an IoT world, the IETF IPv6 over Low power Wireless personal area networks (WPAN), known as the 6LoWPAN working group, was started in 2007, with the goal of specifying an adaptation layer enabling the transmission of IPv6 packets over low-energy IEEE 802.15.4 networks [19]. Thus, 6LoWPAN is basically an adaptation layer, which defines a way to transport IP packets over IEEE 802.15.4 [21] link layer communications. This layer consists of specifications for transmitting IPv6 over IEEE 802.15.4 networks. The payload size in IEEE 802.15.4 networks is limited to 127 bytes, whereas the Maximum Transmission Unit (MTU) in IPv6 networks is at least 1280 Bytes, hence the need to define fragmentation and reassembly mechanisms in the context of the adaptation layer. The adaptation mechanisms defined by 6LoWPAN include the introduction of headers that support packet compression, fragmentation, and reassembly operations. Though RFC 4944 [19] clearly states the need of adapting appropriate security mechanisms for 6LoWPAN, there is currently no security standard protocol defined specifically for the 6LoWPAN layer. Some of the research challenges in 6LoWPAN are security against packet fragmentation attacks, adoption of IPSec/IKE protocols and design of lightweight key management mechanisms. As previously discussed, key management in this context is, in reality, a cross-layer security aspect. Nevertheless, there are a few proposals towards the usage of compressed IPSec [24] in 6LoWPAN environments, while key management (and key bootstrapping in particular) is not covered by such proposals. A simplified version of the IKE protocol [25] has also been proposed for 6LoWPAN communication environments, thus contributing to approach IoT security to the Internet security architecture. Although 6LoWPAN was originally conceived to support IEEE 802.15.4 networks, it is also interesting to note that it is currently being adapted to support other communications technology, particularly Bluetooth Low Energy [26], and this also illustrates the importance of adopting Internet standard communication and security mechanisms to support IoT applications.

C. NETWORK AND ROUTING LAYER

Regarding routing in 6LoWPAN communication environments, it is achieved by the standardized IPv6 Routing Protocol for Low-Power and Lossy Networks (RPL) protocol [27]. RPL aims to reduce energy consumption at the constrained nodes and supports a number of traffic flows like point-to-point, point-to-multipoint or multipoint-to-point communication. This routing framework defines its own security mechanisms, in particular secure versions of the various routing control messages, and various security modes. One mode is the ''preinstalled'' mode, in which sensors joining an existing RPL instance use preinstalled cryptographic keys to secure and process security for RPL messages [4], [28]. An alternative security mode is the ''authenticated'' mode, and in this case nodes may also use preinstalled keys to join an RPL Instance as a leaf. On the other end, joining an authenticated RPL Instance as a router in the RPL topology requires obtaining a key from a Certification Authority authority, but the process by which this key is obtained is currently not defined in the RPL specification [27]. In fact, nor are the mechanisms by which nodes may obtain preinstalled cryptographic keys, and we can clearly see that key bootstrapping is of need in the context of RPL. Key bootstrapping is clearly identified in the RPL specification as an important requirement, where it is noted that a companion specification will be required to detail the mechanisms by which a node obtains the required keys and certificates to bootstrap its secure operations in the context of RPL. Other research challenges in RPL are internal attacks and the design of intrusion detection systems and models to deal with such attacks [4].

D. TRANSPORT AND APPLICATION LAYERS

To enable seamless transportation and support of Internet applications, the IETF's Constrained RESTful Environments (CoRE) working group introduced the Constrained Application Protocol (CoAP) [29], as the de facto standard protocol at the application layer for IoT. CoAP is explicitly designed to meet the requirements of low power and lossy networks in the IoT, namely low overhead, simplicity, multicast support and reduced energy consumption. CoAP uses UDP (User Datagram Protocol), and to address security concerns at the transport layer, the Datagram Transport Layer Security (DTLS) [30] is employed. In practice, DTLS is an adaptation of the Transport Layer Security (TLS) protocol, providing the same security services for applications using UDP. One aspect to note regarding the usage of DTLS is that it is an end-to-end security solution at the transport layer, thus its security is not integrated with the CoAP protocol itself, nor it can support object or message-oriented security approaches. To ensure scalability and efficiency, CoAP also specifies the use of forward and reverse proxies, which in turn requires DTLS to be terminated at the proxy [31]. Therefore, other alternative data object or message-based security methods to ensure security in CoAP communications are still required at the application layer. One possible approach is to integrate security into the CoAP protocol via additional security options, as in [32]. As for security, we note that the current CoAP specification [29] identifies three different and complementary security modes for usage with DTLS: the *PreSharedKey*, *RawPublicKey* and *Certificates* modes, as we proceed to discuss:

- *PreshareKey* mode: In this security mode, devices are pre-programmed with the symmetric cryptographic keys required to support secure communications with other devices or groups of devices. This mode may be thus

appropriate to applications employing devices which are unable to support public-key cryptography, or for which it is convenient to pre-configure security.

- *RawPublicKey* mode: in this security mode, devices use authentication based on public keys, while without being part of a public-key infrastructure. In this scenario, devices are preprogrammed (for example as part of the manufacturing process) with an asymmetric key pair, which can be validated using an out-of-band mechanism, but without having to store and use digital certificates. The identity of the device is obtained from its public key and the device also possesses a list of identities and public keys of the nodes it can communicate with. This security mode is currently defined as mandatory to implement in CoAP.
- *Certificates* mode: in this security mode, authentication is also based on public-keys but for devices that are able to participate in a certification chain for certificate validation purposes. A security infrastructure (a Public Key Infrastructure or PKI) must thus be available, what in practice still represents a challenge, given that most IoT applications are supported by constrained sensing platforms. The devices use an asymmetric key pair stored in a X.509 certificate, which binds the device to an Authority Name. This certificate is also signed by some common trusted root, similarly to the current digital certification architecture of the Internet. The device also holds a list of root trust anchors that can be used for certificate validation purposes.

We may observe that the previous security modes of CoAP do not address how cryptographic keys are bootstrapped at the beginning of the life cycle of the device (please refer to Figure 1), since a concrete key management solution has not been adopted, so far, for Internet communication environments based on 6LoWPAN. Even for the *Certificates* security mode, there are numerous research challenges in what respects the employment of constrained sensing devices in the context of a Public Key Certification Authority, as we discuss throughout the article. Apart from the absence of a specific key management solution, the addressing of these challenges require resource-intensive operations with DTLS. Other issues of note are the lack of a security infrastructure and facilities to support the online verification of the validity of X.509 digital certificates.

IV. A TAXONOMY OF CLASSIFICATION APPROACHES OF KEY BOOTSTRAPPING PROTOCOLS FOR THE IoT BASED ON PUBLIC-KEY CRYPTOGRAPHY

We proceed by presenting a taxonomy of classification approaches of key bootstrapping protocols, which will help us in clarifying and contextualizing the analysis and discussion of the proposals and open research throughout the article. We find it also necessary to analyze related work in this context, as we proceed to discuss.

A. RELATED WORK

A number of key bootstrapping classifications have been proposed previously in the literature, many of which focused on classic Wireless Sensor Networks (WSN) environments, thus without considering its integration with the Internet communications infrastructure. Simplício *et al.* [33] provide a detailed overview of pre-distribution key management schemes in the context of WSN. In this work, the authors cover a broad range of solutions considering pairwise and network-wide schemes in hierarchical networks, probabilistic schemes in distributed networks, matrix-based, polynomial based and combinatorial designs techniques. Such proposals were designed for closed WSN environments, as such not being targeted at IoT standard mechanisms such as those previously discussed and illustrated in Figure 2.

Roman *et al.* [34] discuss the relevance of Public Key Cryptography, pre-shared keys and the link layer key management mechanisms in the IoT context. The authors review only two link layer key management systems (KMS) approaches in this study: the Blom scheme and the polynomial scheme, and conclude that heavyweight mathematical operations of PKC and link layer key management protocols remain a research challenge in IoT. This study did not take into consideration the presence of trusted third parties in key management, which may prove to be a promising approach for widely heterogeneous and scalable IoT networks. Saied [35] propose to classify bootstrapping solutions on the basis of the employed authentication method, the core cryptographic primitive and the key delivery scheme. Nguyen *et al.* [36] propose a taxonomy of key bootstrapping and distribution mechanisms for securing unicast communications in the IoT. The two broad categories proposed are the asymmetric and symmetric key schemes, and the asymmetric schemes are further classified into key transportation and key agreement schemes, on the basis of how key distribution is handled. This study concludes that the need of the hour is the development of efficient security protocols, applicable to constrained sensing devices. This could be achieved by optimizing asymmetric solutions, developing hybrid solutions and adapting the current Internet security protocols for the IoT.

Sarikaya *et al.* [37] classify the available IoT bootstrapping mechanisms into managed, peer to peer or ad-hoc and leap-of-faith/opportunistic methods. For each scenario, the authors analyze the advantages and disadvantages of its deployment in the IoT. The authors also discuss various examples and security considerations related to each method. Das *et al.* [38] proposes a high level taxonomy of IoT security protocols, focusing on major security services as key and identity management, privacy preservation, user and device authentication and access control. The authors conclude that the design of lightweight device authentication and privacy preserving methods, as well as for management authorization in machine-to-machine communications, are much necessary areas of research, in order to secure IoT communications. The authors also advocated the necessity of designing lightweight

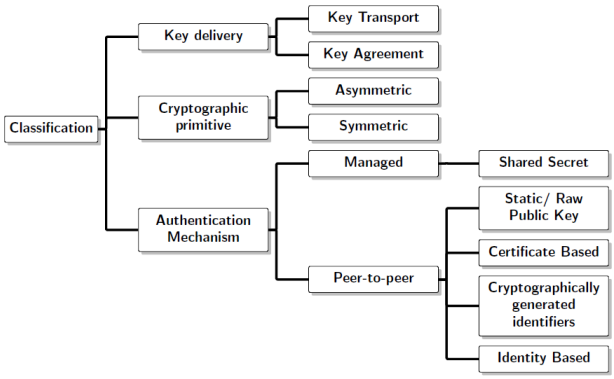

FIGURE 3. Classification approaches of key bootstrapping proposals.

IDS mechanisms to secure IoT networks from internal and external attacks.

As an alternative approach to the previously mentioned works, the taxonomy in this paper focuses on Public-Key (or asymmetric) key distribution schemes for securing unicast communication in the context of IoT applications. We focus in particular on a higher degree of classification, on the basis of the authentication method and the key delivery scheme employed, while also evaluating different types of asymmetric key schemes in detail. For instance, we not only discuss the relevance of PKC as in [34], but also categorize the different approaches based on the key delivery scheme. Also, when compared to works in [36] and [38], we also address out-of-band, implicit and explicit certificate-based authentication mechanisms for key bootstrapping. We finally note that the support of group communications is out of the scope of our discussion. We proceed by analyzing in greater detail the key bootstrapping approaches, in the context of the taxonomy considered to support our discussion, illustrated in Figure 3.

B. CLASSIFICATION APPROACHES

Key bootstrapping (or key establishment/distribution) is one of the fundamental problems of cryptography, whereby a shared secret becomes available to two or more parties for subsequent cryptographic use. Until the 1970s, the only secure way of distributing keys was to use trusted couriers or armed guards. However, this classic strategy had a number of issues, such as its obvious lack of scalability, and the fact that security no longer rests with the key, but with the courier. As previously discussed, all primitives of security as confidentiality, integrity, authentication and non-repudiation require an initial key establishment process, or the secure negotiation of cryptographic keys in the first place. In the particular case of IoT applications, these processes must also be in line with the resources available on constrained devices supporting such applications.

An important concept is that of a protocol, in the context of IoT communications. As for computer systems in general, we define it as a set of rules that two or more communicating entities employ, with the purpose of establishing communications. We may consider two types of protocols, algorithmic and communication protocols [35]. On the one hand, the classical communication protocols define procedures and formats related with packet formation, its transportation, encoding and state transitions. As for algorithmic protocols, they define how communicating entities perform a cryptographic operation. Thus, cryptographic protocols per se are algorithmic protocols, and are usually not related to how data is transported. For example, the IPSec key bootstrapping protocol uses IKE for key management communications, together with the underlying usage of the Diffie-Hellman algorithmic protocol. We also note that, in this survey, we only deal with key bootstrapping cryptographic algorithmic protocols.

Key bootstrapping protocols can be classified along four different approaches, on the basis of the *key delivery scheme* employed, of the underlying *cryptographic primitive* and of the employed *authentication method*. Our subsequent discussion follows this classification, which is encompassed by the taxonomy illustrated in Figure 3.

1) ON THE BASIS OF THE KEY DELIVERY METHOD

In this context, key bootstrapping is broadly classified into two categories: key transport and key agreement, as we proceed to discuss.

- Key transport mechanisms: In this approach, one party creates or obtains a secret value, which is securely transferred to the other party(ies). The secure transfer of that secret value may involve, in practice, the usage of an out-of-band communication medium or its pre-deployment. The resulting secret key is either used directly or derived as some function of the transferred secret. The key transport can a be one-pass, a two-pass or a server-assisted key exchange [35]. In a one pass key exchange, only one peer creates and sends the secret value to other peer. In a two-pass key exchange, both the peers create their own secret values and transfer it to the corresponding partner, and the final key is computed by both peers using the shared values passed through a key generation function. In a server-assisted key transport approach, either the server distributes securely the shared secret key to both peers, or one peer sends the secret value to a server, which subsequently forwards it to the other peer. In such scenarios, the server acts as the KDC (Key Distribution Center) in the former case, and as a KTC (Key Translation Center) in the later.
  Key transport protocols are based on both symmetric, as well as asymmetric, cryptographic algorithms. Transport protocols, based on symmetric cryptography, may or may not require the usage of a trusted server [39], as is the case of Kerberos [40], Shamir's no-key protocol [39], Needham-Schroeder shared-key protocol [41] and Otway-Rees [42], among others. On the other hand, key transport based on public-key cryptography involves one party choosing a symmetric

key and transferring this key to the other party, with such communications being secured using the other party's encryption public key. RSA and ECC are well-known examples of asymmetric primitives employed in the context of transport protocols based on public key encryption.

- Key agreement mechanisms: Key agreement mechanisms are those in which a shared secret is derived by two (or more) parties, as a function of the information contributed by both parties, such that no one is able to predetermine the resulting value. Such mechanisms are resistant to eavesdropping attacks, and are primarily based on the Diffie-Hellman key exchange algorithm [10]. A number of variants of this algorithm have been proposed in the literature, to counter its inherent weaknesses, in particular its vulnerability against man-in-the middle attacks, the lack of authentication of public parameters and long key sizes. Other examples of key agreement protocols include the Station-to-Station (STS) [11], Blake-Wilson, Johnson and Menezes [43], and the Menezes-Qu-Vanstone (MQV) [44] protocols.

2) ON THE BASIS OF THE CRYPTOGRAPHIC PRIMITIVE

Another useful classification of key bootstrapping approaches is on the basis of the employed underlying cryptographic primitive approach. Depending on the underlying cryptographic primitive family, key management schemes can be broadly classified as symmetric key schemes and asymmetric key schemes. Key transport and key agreement mechanisms do exist that rely either on symmetric or asymmetric cryptography [39], as we proceed to discuss:

- Symmetric Key Pre-distribution Schemes: In such schemes, communicating parties share a common secret key and then encrypt or decrypt the messages exchanged using that key. Such schemes are also known as shared key, single key or secret key schemes, and assume that the communicating parties initially share common credentials, which can be a symmetric key, together with some random bytes, flashed into the node before its deployment. As a symmetric key is assumed to be used for communications only with intended users, such schemes ensure implicit authentication in communication. These schemes may also deploy a server or a key distribution center to distribute the keys to nodes, and symmetric schemes provide low computation overhead, which is suitable for constrained sensing devices as in the IoT. Nevertheless, they present their own disadvantages, such as the memory required for storing keys, its low scalability, high communication overhead and vulnerability to node capture attacks. Of course, the usage of symmetric (or secret) cryptographic keys raises the problem on how to securely pre-configure or transmit such keying material in the first place. Also, these schemes cannot achieve non-repudiation, thus the origin of the messages cannot be verified.
- Asymmetric Key Schemes: Asymmetric key schemes are based on the usage of Public Key Cryptography (PKC), and employ two types of keys: public keys and private keys. The public key, as its name implies, is known to all communicating devices, although the private key needs to be kept secret to each communicating entity in the network. The two keys are mathematically related, in such a way that deriving a private key from the corresponding public key is computationally infeasible. The mathematical relation between the two keys is related with expensive mathematical operations like modulus, exponentiation and prime factorization and, due to the fact that such mathematical operations are computationally and energy demanding, they are not usually employed for bulk data encryption, in alternative being used to encrypt smaller key-establishment related communications.

  Such key schemes are called asymmetric key schemes, and not only they ensure confidentiality, but also have the ability to generate digital signatures that support authentication, non-repudiation and integrity. Traditionally, in asymmetric key schemes, messages were first digitally signed and then encrypted. This order has less efficiency and more summation cost and thus, in 1997, a relatively new public key primitive known as Signcryption [45] was introduced. Signcryption performs the functions of digital signature and encryption simultaneously, in single logical step, thereby reducing computation and communication costs. We also note that signcryption is relevant and still effective, as it also motivates recent approaches to security and key management for the IoT [36], [46], [47]. Asymmetric key schemes employ asymmetric algorithms such as RSA, DSA and ECC, and are widely deployed in the conventional Internet. Although asymmetric key schemes have low memory requirements, high scalability and resilience to attacks, they employ computationally intensive operations which increase the energy consumption and computation cost in the context of IoT applications.

Although asymmetric cryptography will certainly play an important part in the future of IoT security, particularly in the context of security bootstrapping and related authentication procedures, its computational and energetic impact still represents some challenges, as we discuss throughout the survey. The public keys must be authenticated to ensure that a public key actually belongs to a particular user or device and, as such, sensing and actuating IoT devices need to be part of a suitable authentication mechanism such as a PKI, and this motivates one of the current research and engineering challenges in materializing security in the IoT.

3) ON THE BASIS OF THE AUTHENTICATION MECHANISM

Users and devices need to be properly authenticated by binding keying material with the identity of the particular entity or device, therefore the authentication mechanism employed can also be used to classify key bootstrapping strategies.

TABLE 1. Classification of key bootstrapping protocols according to key delivery mechanism and authentication mechanism.

			Key Delivery Mechanism	
			Key Transport	Key Agreement
Authentication Mechanism	RPK - Authenticated Pre-distibution		Rabin's Scheme [51]	TFTP-DHKE [52], ECDH-HIP-DEX [53], [54], HIP-BEX [55]
	RPK- Out of band Authentication		-	HIP-DEX [56], ECC/HTTP [57]
	Certificate Based	Explicit / PKI	DTLS/RSA [13], [58], 6LoWPAN [59] DTLS/ECC [60]	DTLS/ECDHE [61]–[63], Compressed IKEv2 [25] IKEv2 [64]
		Implicit	-	DTLS/ECQV/ECDH [65], [66] [67], [68] [69], [70] iSMQV [71], IEEE 802.15.4/CGA/ECDH [72]
	Identity-based		[73], [74], [75], [76], [46], [77]	[78], [79]
	Self-Certified		-	ECC-SCKM [80]
	Certificate-less		MIKEY-ECC [16]	CL-EKM [81]

This classification is based on the initial trust establishment between devices, through authentication of the keys involved in securing the communications. Some algorithmic protocols like one-pass key transport ensure implicit authentication, while protocol as the basic Diffie-Hellman algorithm does not provide authentication, and the origin of its values must be authenticated through signatures or hashes. As with key establishment protocols, authentication mechanisms also rely on either symmetric or asymmetric cryptographic primitives. Symmetric cryptography-based authentication methods rely on pre-established authentication credentials or pre-shared keys, and are often known as managed methods, and those which do not are identified as peer-to-peer or asymmetric cryptography-based methods. Managed methods typically use centralized servers for authentication, whereas peer-to-peer methods usually use an out-of-band (OOB) communication channel, in order to ensure dynamic authentication of the communicating parties [37]. Other methods of asymmetric authentication include identity-based authentication, PKI or Certificate-based authentication and cryptographically generated identifiers [35], which we analyze later in the article.

V. SURVEY OF KEY BOOTSTRAPPING PROTOCOLS FOR THE IoT BASED ON PUBLIC KEY SCHEMES

We proceed our discussion by analyzing the existing proposals targeting key bootstrapping solutions for the IoT, based on public-key schemes. As we verify in our discussion, this involves approaching the eradication of the complexity of asymmetric key management, while without comprising the security level. As shown in Table 1, we classify the applicable proposals on the basis of the authentication (raw public keys, certificate-based and identity-based) and key delivery mechanisms employed.

As already discussed, key transport is a key establishment technique whereby one party creates, or obtains, a shared secret key, which then is securely transferred to the other party(ies) involved in the communications. In the case of key transport, most of these proposals have revisited the security protocols of IKE and HIP, adapting such approaches to the IoT by adopting solutions such as DTLS. In order to achieve the best trade-off between the level of security, the required memory and the computation overhead, researchers propose lightweight versions of such protocols. On the other hand, key agreement protocols in the IoT produce symmetric keys as output, with the resultant symmetric key used to secure data communications. These schemes may or may not use public key signatures to authenticate the communicating parties involved in the communication, and most of such protocols are based on the Diffie-Hellman key exchange [10]. However, in IoT, DH-based solutions are usually considered to be expensive, due to the large bit size of the parameters involved in the computations. In this context, variants of DH [48] based on ECC (e.g. ECDH) have been proposed, which can achieve the same level of security, although using smaller keys. Based on the DH scheme, the MQV protocol [49] is also an authenticated key agreement protocol and, similarly to other authenticated DH variants, ensures security against MITM attacks [50]. We proceed by analyzing in detail the various proposals for key bootstrapping protocols for the IoT based on PKC, starting with raw public key schemes.

A. RAW PUBLIC KEY (RPK) SCHEMES

Raw public keys (RPK) are public keys that are pre-deployed on the devices, either in off-line mode, or through some out-of-band mechanism. Accordingly, the authentication is ensured off-line, or when an out-of-band mechanism binds the public key to the entity/identity presented by the key. To reduce the burden of certificates on resource-constrained devices, and also to increase its efficiency, the use of raw public keys for TLS and DTLS has been standardized by the IETF [82]. Even though these schemes require less message exchanges than certificates and identities, they can be used only for small network scenarios, in the context of which the public key of each node is known to all other nodes beforehand. Similar to RSA, Rabin's cryptosystem [83] is based on the integer factorization problem, while unlike RSA and other asymmetric cryptosystems such as ECC, it possesses characteristic of computational asymmetry. In Rabin's scheme, encryption is relatively lightweight than decryption, thus making it suitable for the IoT, where devices are constrained but a gateway is usually employed without such restrictions, to aid in security-related tasks. As previously discussed, raw public keys are deemed mandatory in CoAP security using DTLS. On the basis of the authentication technique, we may characterize schemes based on raw public keys as those using out-of-band authentication and pre-distribution, as we discuss next.

1) RPK-AUTHENTICATED PRE-DISTRIBUTION

This approach is based on the assumption that public keys have been authentically distributed beforehand, either off-line or on-line, and are as such ready to be used to secure communications in the context of a given IoT deployment. In off-line mode, public/private key pairs are securely preloaded or preprogrammed in the devices at the time of manufacturing. However, they are transmitted upon request from a public authority in on-line mode. The public/private key pair thus

obtained can be directly used to either secure the communication (e.g. in RSA, ECC), to initiate a session key exchange (e.g. in DH, ECDH) or to generate digital signatures (e.g. in ECDSA). The device identity is derived from a public key and a list of identities with the public keys of the nodes it wants to communicate with and, in this context, the public key binded to a device's identity is assumed to be authentic.

To enable key transport in the IoT, one of the first approaches using Rabin's cryptosystem [83] was proposed in [51]. In this work, the sensor nodes run self-configuration mechanisms to act as service nodes, and subsequently responsible for generating the keying material required to ensure secure communication between worker nodes. The service node broadcasts its unique ID and its public key to all worker nodes in its range, and next the worker node encrypts the session key with the service node's public key, using the Rabin's cryptosystem. However, this scheme is computationally expensive, since the relatively heavier decryption operation of the Rabin cryptosystem is performed by service sensor nodes.

To ensure key agreement for IoT applications, Isa *et al.* [52] employ a DH key exchange (without any modification or optimization) and propose a security enhancement to the Trivial File Transfer Protocol (TFTP) for smart IoT environments. Hummen *et al.* [53] analyzed the effect of public key management schemes in the DTLS, HIP-DEX and minimal IKEv2 protocols. In this work, the authors identify three major challenges in the adoption of such schemes for IoT, namely the expensive public key operations, the increased risk of DoS attacks and the fixed timeout retransmission mechanisms which are part of the protocol handshake. To address such challenges, the authors implement and evaluate lightweight extensions to the static ECDH keys used in the HIP-DEX protocol, which they verify to have a marginal overhead of 0.2 KB of RAM and 5.6 KB of ROM. These extensions include the session resumption mechanism, puzzle-based DoS protection and a refined retransmission mechanism. The authors argue that the same extensions could also be generalized for DTLS and minimal IKEv2.

Hummen *et al.* [54] proposed Slimfit, a compressed HIP DEX layer used to modify the packet structure of HIP DEX, by compressing redundant information and omitting irrelevant information, while at the same time retaining the general semantics of the protocol to ensure compatibility. Other advantage of the proposed scheme is the verified reduced packet transmissions and fragmentations. The authors also discuss related security considerations and mechanisms, and conclude that the integration of Slimfit with the network stack may be beneficial to secure IoT applications using the HIP or HIP DEX protocols.

Sahraoui and Bilami [55] propose the Compressed and Distributed HIP (CD-HIP) model, based on the HIP header compression scheme, as well as on an adapted key distribution scheme for HIP Base EXchange (HIP-BEX). The authors combined both compression and distribution models to develop CD-HIP, and claim that CD-HIP is energy efficient and offers good compatibility with the standard HIP protocol.

2) RPK OUT-OF-BAND AUTHENTICATION

Out-of-band authentication mechanisms are appropriate to network scenarios and devices which do not possess pre-shared symmetric secrets, or the authenticated public keys of other devices, and also that are not able to efficiently be part of a PKI. The main aim of out-of-and authentication is to exchange some limited amount of confidential and authenticated information between pairing devices, and to use this information to authenticate the public key over insecure wireless channel [84]. Thus, user or device authentication is performed over an alternative channel or network, rather than via the primary communication channel. One such approach is to employ DNSSEC for the purpose of supporting authentication, when public keys are obtained through DNS-Based Authentication of Named Entities (DANE) [85]. Other out of band mechanisms may be used to accomplish device and group device pairing, as discussed in [84]. The specifications in [82] define extensions for the exchange of raw public keys in DTLS [30]. Meca *et al.* [56] propose a security architecture based on HIP and AMIKEY [86], to support secure network associations and key management. This architecture assumes that devices are pre-configured with identifiers, i.e. using HIP-DEX public keys, and that each IoT network domain is part of a central authority responsible for the management of that domain. Out-of-band authentication is based on the knowledge of a symmetric key at layer two of the IoT domain. Then, the pairwise key obtained from the HIP handshake is used as a master key in AMIKEY, and is continuously updated and refreshed using the crypto-session bundle. The authors conclude that their proposed architecture requires fewer message exchanges, and thus fewer bytes exchanged between the initiator and the responder. In [57], a key agreement approach using ECC-based public/private keys is proposed, where the authentication between the IoT devices and the cloud server is performed using HTTP cookies. The ECC-based scheme employed to derive the symmetric session key is lightweight, but is fails to achieve mutual authentication and is also vulnerable to device identity spoofing attacks.

B. CERTIFICATE-BASED SCHEMES

From a security standpoint, it is well known that one of the best approaches to authenticate public keys is to have the various entities participating in a PKI (Public Key Infrastructure). A PKI defines, in practice, the set of policies and procedures to manage public key encryption and also the creation, distribution, management, storage and revocation of digital certificates. A PKI ensures authentication of public keys of users and devices, by binding them with its identities. In a PKI, a third trusted party, known as the Certificate Authority (CA) holds the responsibility of registration and issuance of certificates to the various entities, while the Registration Authority (RA) ensures its valid and correct registration. The third component in a PKI is the repository that stores

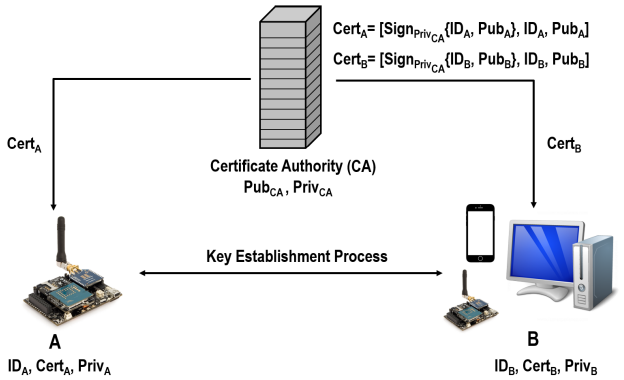

FIGURE 4. Mutual authentication with explicit certificates in a PKI.

certificates and Certificate Revocation Lists (CRL). The digital certificates issued by the CA are verified by a chain of trust, and in order to map the services of a PKI in a given IoT network, the root node can act as root CA responsible by the registration, issuance, storage and revocation operations. A certificate has three major constituents: the identification data, a public key and digital signature that binds the public key to the identify of the user. We note also that certificates may be managed implicitly or explicitly, as we proceed to discuss.

1) EXPLICIT CERTIFICATES/PKI

Explicit or conventional certificates are managed and signed by a trusted third party (a CA), and any entity in the network is able to validate the certificate by verifying the signature of the CA contained in it. Once the CA's signature on the certificate is explicitly verified, the users are assured that the public key actually binds to the claimed user's identity. These certificates can be preloaded into the devices, or directly acquired on request from the CA. This process is illustrated in Figure 4.

We must note that X.509 is the most widely accepted certificate standard, which specifies the format of public key certificates, and X.509 certificates are used in a number of protocols such as TLS/SSL, HTTPS, S/MIME, EAP-TLS and IPSec/IKE, among others. Explicit certificates have been used for key transport [13], [58]–[60] as well as for key agreement [25], [61]–[64] for the IoT. We must however note that the employment of digital certificates by IoT constrained devices is quite expensive, particularly considering the scarcity of resources such as computational power, storage space and energy in most IoT sensing and actuating platforms. Due to this fact, various research proposals have been published which target the usage of lightweight explicit certificates to secure IoT applications, as we proceed to discuss. These works focus on the compression of protocol headers, delegation of computational intensive operations, secret sharing theorems, using specialized certificates or certificate profiles.

Initial research on PKI for key management in the IoT focused on the compression of protocol headers [25], [61], [63], starting in 2012 when Raza *et al.* [61] deployed PKI at the DTLS layer in IoT. In this work, the authors analyzed the headers of DTLS and found they are too long to fit in a single IEEE 802.15.4 packet. Thus, they proposed a novel 6LoWPAN compression for DTLS, in which they argue to reduce the size of DTLS headers. As a result, the authors achieve a 75% savings in the number of bits of the DTLS handshake header. The DTLS handshake is responsible for automatic key management at the transport layer, authenticating the server and client using a PKI and ensuring end to end security. However, the authors neither evaluate the PKI nor explain how public keys are to be distributed between entities. Raza *et al.* [25] propose the compression of IKEv2 payloads using 6LoWPAN NHC. In order to establish a security association at the IEEE 802.15.4 link layer, this study proposes a new protocol identifier for IKEv2 security association payloads. Besides, the authors also plan to replace RSA with ECC for automatic key exchange in IKEv2. However, the authors evaluate the proposed schemes only theoretically and, in fact, an experimental evaluation is missing from this study, as is an evaluation on the impact of using digital certificates. Later, Raza *et al.* [63] propose and implement compression for DTLS on 6LoWPAN communication environments [61]. The authors propose DTLS header compression, along the same strategy proposed in [62]. The proposed compression mechanisms reduces the message size and thus saves energy in communications. Additionally, even if 6LowPAN is susceptible to fragmentation attack, the scheme is more secure than conventional DTLS, because it reduces fragmentation. Though this work is based on pre-shared keys, the authors propose its implementation with certificates in the context of DTLS. Also, this study does not suggests backward compatibility of header compression with the conventional DTLS protocol.

Despite the existence of the previous approaches, we must note that the first fully implemented two-way authentication solution using certificates was proposed by Kothmayr *et al.* [13]. In this work, a fully authenticated DTLS handshake based on the exchange of X.509 certificates with 2048-bit RSA keys has been proposed. In this work the authors implement RSA on Opal sensor nodes, a device powered with integrated Trusted Platform Module (TPM), employed to support the generation and storage of RSA keys, as well as hardware support for the RSA algorithm. Also, in platforms without TPM, the authors found that ECC implementations require significantly less resources than when using RSA. The authors analyze the resource requirements of a DTLS implementation using ECC, and integrated RSA and ECC-based DTLS with the TinyIPFIX [87] protocol for usage in a building automation application scenario, but however verified that the proposed implementation still requires more resources than with less resourceful devices.

Other works focused on delegating the heavy PKI operations to devices with less resource constraints, in particular an edge router or gateway device [59], [62]. To reduce the overhead of certificate based DTLS handshake, Hummen *et al.* [62] propose three design ideas. These ideas

involved certificate pre-validation, session resumption and the delegation of costly handshake operations to a more capable gateway device. In certificate pre-validation and handshake delegation, all the certificate-related procedures are moved to the gateway device. In session resumption, the key idea is to perform expensive operations only once, during the handshake. The authors propose this idea for DTLS, but they claim that this could also be extended to alternative protocols, namely IKEv2 and HIP. The proposal is evaluated and analyzed considering the communications, computation and memory overhead, and verified to require slightly more memory for the support of the DTLS protocol. Also, this work did not specify how and when the certificates are obtained from a central CA. Misra *et al.* [59] proposes an integration model for PKI and 6LoWPAN, without changing the conventional PKI primitives. The proposed model delegates a major portion of the heavy PKI computations to the edge router, and the responsibilities of this router include gathering the public keys of all nodes in the 6LoWPAN network from the CA, maintaining the certificates in a local database and synchronizing with CA server over the wired IPv6 infrastructure. The authors perform the evaluation of the proposed model using the Perytons Protocol Analyzer, but the drawback of this model is that its requirements are applicable only to industrial and control 6LoWPAN networks. On the other hand, if compromised, the edge router is in reality a single point of failure.

Specialized compact public key certificates like self-descriptive and non-self-descriptive card verifiable (CV) certificates [88] for IoT were deployed in [58]. This work discusses the applicability of X.509 v3 certificates to the requirements of the IoT, and identifies and attributes certificates along with self-descriptive and non-self-descriptive CV certificates. Among such certificates, the authors suggest the usage of combined identity/attribute certificates, which consists only of a single root CA, and with the lifetime required according to the operational lifespan of the device. However, the authors did not discuss revocation nor the verification of the status of certificates, from an IoT perspective.

It is evident from our previous discussion that, so far, proposals do not address modifications to the X.509 format, with only a few designing compression methods [25], [63] to adapt the standard to the IoT. However, it is possible to compress or optimize such certificates for IoT networks. One such scheme to compress X.509 certificates has been proposed by McGrew and Pritikin [89], and in this work the authors defined a compact format for X.509 certificates, together with methods to translate between the standard and compressed formats. To accomplish compression and decompression of digital certificates, the authors employ the DEFLATE algorithm [90] with a pre-configured dictionary. Edgecombe [91] extended the CXF certificate by developing a new dictionary from a sample of 1,000 certificates. Though these works discuss certificate compression, their direct application in IoT networks is not addressed. Recently, Forsby *et al.* [60] developed an X.509 profile for the IoT, by excluding non-required fields from the certificate, and compressing the fields that are indeed employed, using the CBOR encoding scheme. The proposed profiling is compatible with the X.509 standard, and also consumes less memory and energy, when compared to its uncompressed counterpart. Similar works have been proposed in [92], where the authors reduce the size of typical X.509 certificates by nearly 30%.

Another work [64] tailored the Chinese Remainder Theorem (CRT) for secret sharing, in the context of certificate-based Diffie-Hellman key exchange in IKEv2. The authors propose the Cooperative Key Exchange System (CKES) based on the secret sharing theorem, i.e. the (CRT). In this system, an initiator constrained device requests a highly trusted node in the network to initiate a key exchange with another node. The initiator node also mentions the number of collaborative nodes from its cluster, which are required to support heavyweight cryptographic operations. The NS2 simulator was used in this proposal to evaluate the proposed system, in what respects the required energy consumption. While the computation energy decreased with CKES, the communication costs increased, given that at the end more messages are exchanged between devices.

2) IMPLICIT CERTIFICATES

Implicit certificates are another variant of public key certificates, where all the components of certificate, i.e. identification data, public key and digital signatures, are superimposed on one another, in such a way that the size of the certificate is equal to the size of the public key [93]. When compared to explicit certificates, in the context of which the certificate components are distinct elements, the size of implicit certificates is considerably smaller, because digital signatures are superimposed on the public key. The fact that this type of certificates is called implicit is related with the fact that the public key can be extracted and verified from the signature portion of the digital certificate. In other words, there is no need to explicitly validate the signature of the Certificate Authority, present in the certificate. Instead, users extract the public key from the implicit certificate and use it in the intended operations of signing or key agreement protocols [50]. This makes implicit certificates faster than conventional certificates, and we can consider that such certificates are a promising approach to identify and certify devices, in the context of future IoT applications. Given its smaller footprint, such certificates are also faster to process in constrained sensing devices. Figure 5 illustrates the main differences between explicit and implicit certificates, in what respects its main forming components.

The elliptic curve variant of MQV, the Elliptic Curve Qu-Vanstone (ECQV) scheme, is the most common type of implicit certificates, and is defined in [50]. The process of key bootstrapping in implicit certificates involves two steps or phases, namely the registration phase and the certificate issuance or authentication phase. In the registration phase, the user requests an implicit certificate from the CA, validates the request and responds the user with the implicit certificate.

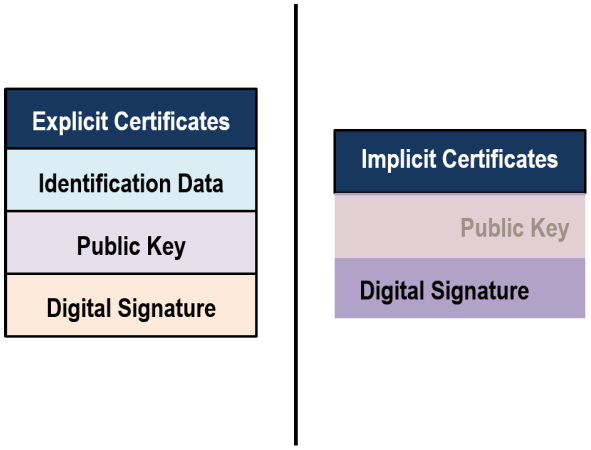

FIGURE 5. Explicit v/s implicit certificates.

The user then computes its public key from the implicit certificate, and correspondingly generates a public/private key pair. To validate the implicit certificate request in the registration phase, works usually assume the presence of shared secret key either network-wise [65]–[68], [70] or pairwise [72] and authenticated identities [69], [71]. In the second phase, the communicating entities derive a session key in order to apply security to the communication, probably employing a symmetric algorithm. Thus, implicit certificates are effectively key agreement mechanisms.

The utilization of implicit certificates in the IoT was introduced in [65] and [66]. In these works, the authors propose a two-phase authentication protocol, consisting of certificate registration and authentication, for IoT networks secured with DTLS. The nodes, as well as the trust root (thus, the Certification Authority), are pre-configured with common secret keys and tagged with authentic identities. However, we verify that such design options are not totally compliant with the DTLS standard, and consequently the authors extended their proposal for the usage of implicit certificates in [67] by proposing PAuthKey (Pervasive Authentication protocol and Key establishment) protocol, in this case relying in the security available at the link-layer with IEEE 802.15.4. The network architecture of PAuthKey includes end users (applications, human users or virtual entities) and heterogeneous edge devices (sensor nodes). For communications between edge devices, some nodes were designated as cluster heads, and perform the functions of a trusted certificate authority. In this proposal, the registration phase uses 6LoWPAN identities and a pre-shared network key. This protocol is able to retrieve the session key with twelve message exchanges (six each in registration and authentication phase).

The assumption of availability of pre-deployed keys was also considered by in [68], which proposes a KMS for IEEE 802.15.4 integrating ECQV implicit certificates with the Elliptic Curve Diffie-Hellman Key Exchange (ECDKE) algorithm. The proposed protocol is able to reduce the number of messages exchanged to four, and also reduces the memory footprint of the security implementation, on constrained sensing device. To achieve this goals, the authors propose a number of optimizations on modular arithmetic, scalar multiplication and task list depth, and the protocol assumes that implicit certificates are preloaded in each device during the device bootstrapping phase. Sciancalepore *et al.* [70] extended their work in [68], and implement the protocol on hardware using modern IoT devices such as the OpenMote platform [94]. The proposed protocol ensures authenticated key derivation and exchange at the application layer, protection against replay attacks and minimal airtime consumption. The authors concluded that the proposed protocol savings in airtime consumption are up to 86.7%.

In 2016, Simplicio *et al.* [71] leveraged implicit certificate in [50] for public key authentication, followed by SMQV (Strengthened MQV) [95] for symmetric key exchange between any pair of nodes. The authors argue that the combination of the low communication overhead of implicit certificates, together with the high security of the SMQV protocol, is able to achieve energy-efficient and escrow-free authentication in the context of key agreement. The authors reduce the communication overhead to only two messages each, in the certificate issuance and authenticated key agreement phases. Later, Ha *et al.* [69] designed and implemented the ECQV certificate, which they use to authenticate the key exchange in the context of the DTLS protocol. A fog-based network architecture is proposed and deployed, in order to test the proposed work, in the context of which the IoT devices and gateways communicate with a CA via the communications infrastructure domain. To achieve ECQV implicit certificate-based key agreement, a new cipher suite for DTLS is defined to convey the message of ECQV generation. Though this schemes is extremely time-efficient, the certificate registration phase requires pre-shared keys for authentication and integrity.

As previously discussed, so far the research proposals assume that the identities are authenticated or the identity of the requester of the implicit certificates is neither being spoofed nor impersonated. Also, device identifiers are not secured during certificate generation and issuance. Thus, we verify that the initial security bootstrapping for establishing a security association between the requester and the CA is not part of such proposals. With this in mind, Park [72] catered to these weaknesses by introducing individual keys for authentication, instead of a single network-shared key, used by all devices and the CA. This individual key is derived from a secure ECDH key exchange, integrated by the authors in the IEEE 802.15.4 join protocol. Since IEEE 802.15.4 uses 64-bit MAC addresses to identify devices, the authors authenticate the ECDH key exchange with cryptographically generated addresses (CGA). CGA addresses are interface identifiers for IPv6 addresses, generated from the cryptographic hash of the public key. Authentication based on the usage of CGA addresses follows the assumption that authenticated identifiers can be obtained from the public key. Thus, the authors employ CGA for device identifiers and ECDH for key exchange. The only assumption taken into consideration is that all network devices must know each

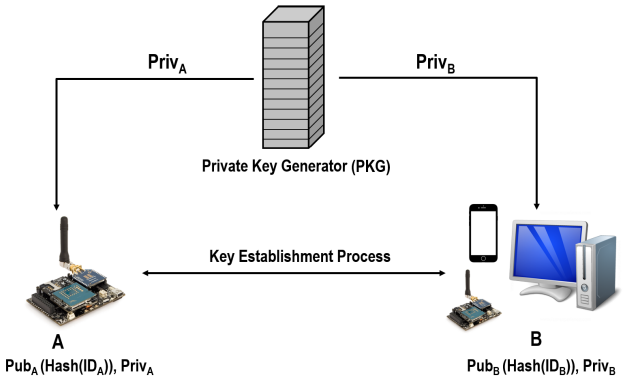

FIGURE 6. Operation of identity-based schemes.

other's MAC address in advance, which are used to identify devices.

C. IDENTITY-BASED SCHEMES

The concept of Identity-Based Cryptography (IBC) was introduced by Adi Shamir in 1984 [96], which proposes to employ user identity attributes, as its phone numbers or email address, in place of digital certificates for the purpose of verifying signatures. This idea was further extended by Boneh and Franklin [97], who introduced IBC based on bilinear pairings on elliptic curves. Currently, IBC is also described in the well-known IEEE 1363.3 standard [98]. Unlike certificate-based schemes, there is no need to generate, manage or revoke user's certificates in IBC-based schemes and, additionally, it requires zero configuration at the message recipient's end. However, IBC relies on a trusted third party known as the Private Key Generator (PKG). As shown in Figure 6, the PKG generates the master private key pair before operation, and next this key is sent to the respective entity. However, we may note that identity-based schemes are vulnerable to key escrow attacks, since the PKG knows the private keys of all its users and, to overcome such attacks, So *et al.* [73] propose to employ distributed key generating servers or short-lived master keys. The authors applied the proposed identity based signcryption scheme in smart grids, to provide zero-configuration encryption and authentication solutions.

Initially, IBC was implemented using RSA, but ECC and EIGamal have also been explored for the purpose of implementation. In constrained IoT communication environments, IBC based on ECC has been extensively studied [74], given that it provides a less expensive alternative to RSA for implementation on constrained devices. Li and Xiong [75] propose an heterogeneous on-line and off-line signcryption scheme (HOOSC), to be employed between IoT sensor nodes and Internet hosts, which supports end-to-end integrity, confidentiality, non-repudiation and authentication. In this scheme, the authors employ bilinear Diffie-Hellman inversion, which enables secure communications between sensor nodes implementing IBC and Internet hosts implementing the PKI. To reduce the required processing time, signcryption is performed in two phases, the off-line phase, in the context of which heavy computations are performed without the knowledge of the message, and the on-line phase, where light computations are performed when the message becomes available.

One Time Password (OTP) authentication based on Lamport's OTP algorithm and IBC-ECC was proposed in [76]. In this work, the authentication is performed by the PKG in the cloud, which shares the OTP with both the requesting application and the device. To make the scheme lightweight and robust to attacks, the authors replace the hash function in the Lamport's OTP algorithm by their proposed function, based on the Identity-based encryption (IBE) scheme. An extension to the work in [75] was proposed by Li *et al.* in [46] and, in addition to the security services in [75], this heterogeneous ring signcryption scheme also envisages privacy of the sensor nodes. The authors use the random oracle model, to prove that the proposed scheme satisfies anonymity, unforgeability and confidentiality. Besides, they also compare the proposed scheme with some of the existing schemes on basis of two parameters, namely energy consumption and computational time, which are evaluated as the number of identities increases. The authors conclude that, in key distribution, public key cryptography is simpler than the symmetric counterpart, and also that it is able to guarantee the properties of non-repudiation and anonymity. The infrastructure also includes a key generation center, which issues the public/private key pair to all the stakeholders of the system based on their identities, using the Menezes Vanstone Cryptosystem [99]. To ensure non-repudiation, the augmented feature in this system is the creation and delegation of certificates, by the key generation center, to the various stakeholders. This exchange of identities, certificates and keys is assumed to be performed using a secure channel.

An augmentation of the traditional identity-based scheme was proposed in [77], and in this work the authors propose an intelligent and secure health monitoring system using IoT-enabled sensors and cloud services. In this scheme, gateways often act as intermediaries to mutually authenticate users to sensor devices, and vice versa. Here, users are authenticated using an out-of-band mechanism, typically with biometrics or smart-card based authentication. Also, in this model devices employ public keys for authentication, pre-deployed by the gateways.

Jiang *et al.* [78] propose a three factor authentication and key agreement scheme, based on Rabin's cryptosystem, for Internet-integrated WSN using smart cards and biometrics. This scheme was designed to enable a remote user to access sensor node data through a gateway, and may be applied to critical applications requiring real-time sensor node information, e.g. in health-care and surveillance. More recently, Saeed *et al.* [79] introduced identity-based authenticated key agreement between client sensor nodes and cloud servers. In this work, the base station hosts the PKG supporting ID-based schemes, which issue the private-public key pairs and other system parameters to the communicating entities

in the network based on its identities. On the other hand, the cloud and the sensor nodes create its own Diffie-Hellman ephemeral keys using random integers and ECC-based curve multiplication. These ephemeral keys are signed using a one way hash function by the sensor and the cloud application, and then exchanged with the other party. After verification of each other's ephemeral keys, the communicating entities conclusively derive the shared key. This work guarantees perfect forward secrecy, key confidentiality, key control and scalability, which are absent from other similar works. Also, it relieves the PKG from complete key dependence because, even if the adversary compromises the PKG, it is not able to derive the shared key.

D. SELF-CERTIFIED SCHEMES

Introduced by Girault [100], self-certified public keys represent another alternative to traditional certificate-based key exchanges. The idea of self-certified keys lies behind the fact that public key are computed by both the authority and the user, so that the certificate is effectively embedded in the public key itself. Thus, there is no need to attach a separate certificate with the public key in order to authenticate it. In other words, self-certified keys have an implicit certificate that can only be generated by the trusted authority [42]. Thus, self-certified keys are advantageous over traditional PKI, as they reduce the storage and computation requirements to support public keys, while designating the user to choose its private key and keeping it anonymous to the authority. The later characteristic makes self-certified certificates also advantageous over ID-based schemes. One of the disadvantages of self-certified keys over traditional PKI is their repudiability. Thus, if the verification of digital certificate fails using a self-certified public key, it is uncertain whether the public key or signature is incorrect. Another disadvantage is that the security provided by self-certified keys against forgery can be reduced to the security of the underlying signature algorithm. On the other hand, while self-certified keys do not present any structural advantage over traditional PKI approaches, they may support an elegant and efficient strategy to support certification [42]. In self-certified keys, a user chooses its secret, and next computes an integer as a function of that secret, which it sends to a Trusted Third Party (TTP), as Figure 7 illustrates. The TTP combines this integer with the user's identity to generate a witness, and this witness can be an authority's signature or some trapdoor function combining the user' s identity with its public key. This witness is the authenticated public key of that user and, given the authority's public key, witness and identity, other users may derive the public key of a particular user. Further storage requirements are reduced in subsequent work [101], in the context of which the requirement of using the authority's public key to compute public key is also removed. Similarly to implicit certificates, the successive usage of the private key itself verifies, implicitly, the authenticity of the respective public key. Also, the link relating the identity and the public key represents in itself a lightweight certificate.

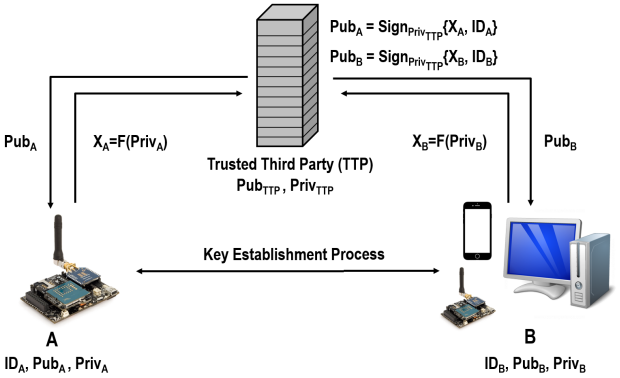

FIGURE 7. Self-certified schemes.

Haripriya and Kulothungan [80] propose an ECC-based self-certified key management scheme (SCKM) for IoT systems. The proposed scheme is based on the concept of zero-knowledge proofs (ZKP), and consists of two phases, the registration phase, and the session phase. In the registration phase, the base station generates the public key of a node and creates a witness for that node, with its own private key. On receiving the public key, the node verifies its private key with the witness received from the base station. In this work, the base station generates the public key of the node without the knowledge of the node's private and master keys, and the technique of zero knowledge proof was successfully implemented. The authors evaluated this scheme using the Cooja simulator, and claim that it is more energy efficient than other certificate-based schemes.

E. CERTIFICATELESS SCHEMES

Certificateless Public Key Cryptography (CL-PKC) is a variant of identity-based cryptography and traditional PKI, and aims to prevent the key escrow problem of IBC while, at the same time, to remove the need to use certificates for key authentication by the PKI. Introduced in [102], CL-PKC schemes implement a special organization of public/private key pairs, where a private key is partially generated by the KGC. This key is identified as a Partial Private Key (PPK). As illustrated in Figure 8, CL-PKC schemes still involve a trusted third party, often identified as a Key Generation Center (KGC). The KGC computes the PPK from the identifier of the entity and the Master Key (MK), and the entity then combines its PPK with some secret in order to compute its actual private key. To calculate the public key, the entity also combines its secret with public parameters of KGC. Therefore, these schemes do not suffer from the key escrow problem, as the full responsibility for the node's private key is taken away from the KGC. This scheme is also not identity-based, because public keys are no longer calculated from identities. In the context of IoT applications, various cryptographic primitives, as signatures [103], encryption [104], key agreement and proxy decryption, have been implemented employing CL-PKC. However, in our discussion we focus our

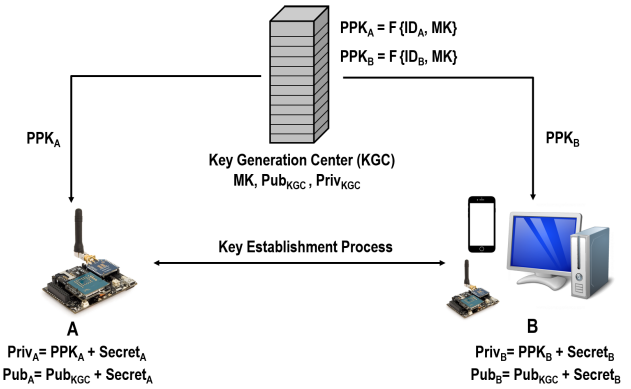

FIGURE 8. Certificateless schemes.

attention on certificateless key agreement protocols for IoT, as they belong in the context of this survey.

A number of CL-PKC proposals have been designed for WSN environments, however only a few of them fit to the characteristics of IoT networks and applications. In 2015, Seo *et al.* [81] proposed a certificateless effective key management (CL-EKM) scheme for dynamic WSN, which can also be applied to IoT networks. This scheme utilizes ECC with 160-bit keys, and ensures authentication via a signcryption scheme. The base station hosts the KDC and assigns a unique identifier, an individual key and a certificateless public/private key pair, to each device. The certificateless public/private key pair is next used to establish a master secret key, which is further used to derive the pairwise session secret key. This scheme supports key updating and revocation, and ensures backward and forward secrecy. Nevertheless, and though the proposed scheme is lightweight, it does not ensure perfect forward secrecy feature in the context of key establishment.

In the context of the IoT, a novel ECC-based signcryption scheme (ECKSS) for MIKEY was proposed in [16]. In this work, the authors postulated two novel certificateless key transport mechanisms for MIKEY, with both mechanisms being PKI independent and based on elliptic curve Korean signcryrtion scheme for enveloping keys. This scheme can be categorized as certificateless, since it relies on binding the public values of the communicating parties with the public keys generated by a trusted key management server. We note that, in spite of the fact that the non-utilization of certificates in CL-EKM wipes out the computational overhead typical of certificate management, the overhead of public key encryption still remains a challenge. It is also worth to make reference that, with the utilization of certificateless schemes, some contrary issues could happen, where users may be not able communicate with other users outside the network, that do use certificates.

VI. ANALYSIS AND DISCUSSION

In this section, we compare the previously identified and analyzed solutions and, for the purpose of our comparison, we base our analysis on the security and functionality features (SFF), the performance evaluation features (PEF) and the applicable IoT communications and security protocols. The performance evaluation features considered are the cost of supporting communications, computations and memory. In Table 2 we classify the reviewed approaches on key management considering such features. We must note that not all works apply directly to IoT protocols, since we also consider, in our analysis throughout the survey, other proposals which may contribute and bring light into how key bootstrapping using asymmetric cryptography may evolve in the context of the IoT security infrastructure.

We note that key establishment protocols give careful consideration on compromised keys and, therefore, we only consider security and functionality features related with the management of compromised keys. Firstly, mutual authentication ensures two-way authentication of sensor nodes with another sensor server node or user, in the context of a given IoT application. Secondly, perfect forward secrecy (PFS) ensures that session keys will not be compromised, even if the long-term keys of the communicating principals are compromised. Thirdly, and the most overlooked feature, we need to consider resilience against key compromise impersonation, in the context of which the compromise of the long-term key of a principal does not allow the adversary to masquerade to that principal as a different principal. Finally, we also consider the resilience to modification and replay attacks. In the category of the performance evaluation features, the communication, computation and storage overhead of the proposed solutions have been considered. The communication cost considers the impact of the security (key boostrapping) mechanisms, related with the number of bytes required in the context of the message exchange, while the computation cost may include the number of operation or comparisons performed or required, to accomplish the key exchange. Lastly, the storage cost considers the additional RAM or ROM requirements to implement the key bootstrapping approaches.

We start by noting that the security and functionality feature of mutual authentication is, in general, well covered by the proposed solutions. As for resilience to replay attacks, it is also well covered, with notable exceptions of some proposals for 6LoWPAN [55] and DTLS/CoAP [53], [54], [63]. We also note that resilience against key compromise impersonation is a security requirement not addressed by various proposals, notably [68], [70] at the link-layer, [55], [59] at the 6LoWPAN adaptation layer and [53], [54], [63] for CoAP with DTLS. On the other hand, we can see that perfect forward secrecy (PFS) is more well supported in such works, and also that not all proposals address resistance against replay and modification attacks. In general, the evaluation or evolution of the previous proposals, with the goal of supporting the previously identified security and functionality features at the various layers of the stack illustrated in Figure 2, constitute opportunities to conduct further research in this area.

Regarding the performance evaluation features of the analyzed works, we may observe that in some works further work

TABLE 2. Comparison of key management schemes based on security & functionality feature (SFF), performance evaluation feature (PEF) and IoT protocols.

Reference	Features								IoT Protocol
	SFF1	SFF2	SFF3	SFF4	SFF5	PEF1	PEF2	PEF3	
[51]	×	×	×	×	×	✓	-	-	-
[52]	×	×	×	×	×	-	-	-	$TFTP^*$
[53], [54]	✓	×	×	×	×	✓	-	✓	DTLS
[55]	✓	✓	×	×	×	✓	✓	-	6LoWPAN
[56]	✓	✓	✓	✓	✓	✓	-	✓	DTLS
[57]	✓	✓	✓	✓	✓	✓	✓	✓	$HTTP^*$
[61]	✓	-	-	-	✓	-	-	-	6LoWPAN/DTLS
[25]	✓	-	-	-	✓	-	-	-	IPSec/IEEE 802.15.4
[13]	✓	-	✓	-	✓	-	-	✓	DTLS
[62]	✓	✓	✓	✓	✓	-	-	-	DTLS
[63]	✓	×	×	×	✓	✓	-	✓	DTLS/CoAP
[59]	✓	-	×	×	×	✓	-	-	6LoWPAN
[58]	-	-	-	-	-	-	-	-	-
[60]	✓	✓	✓	✓	✓	-	-	✓	CoAP/DTLS/6LoWPAN
[64]	✓	✓	✓	✓	✓	✓	✓	-	IPSec/IKEv2
[67]	✓	✓	✓	✓	✓	✓	✓	✓	DTLS/6LoWPAN/IEEE802.15.4
[68]	✓	✓	×	✓	✓	✓	✓	✓	IEEE 802.15.4
[69]	✓	-	-	-	-	-	✓	-	DTLS
[70]	✓	✓	×	✓	✓	✓	✓	✓	IEEE 802.15.4
[71]	✓	✓	✓	✓	✓	-	✓	-	-
[72]	✓	-	-	✓	✓	✓	✓	✓	IEEE 802.15.4
[73]	✓	✓	-	✓	×	×	×	×	-
[74]	-	-	-	-	-	-	-	-	-
[75]	-	✓	✓	✓	✓	×	×	×	-
[76]	-	✓	-	✓	✓	✓	✓	✓	-
[46]	-	✓	-	✓	✓	×	×	×	-
[77]	✓	✓	-	✓	✓	✓	✓	✓	-
[78]	✓	✓	✓	✓	✓	✓	✓	✓	6LoWPAN
[80]	✓	-	-	✓	✓	✓	✓	✓	-
[81]	✓	✓	-	✓	×	✓	✓	✓	-
[16]	-	✓	-	✓	×	×	✓	✓	$MIKEY^*$

Note: SFF1: Mutual Authentication; SFF2: Perfect Forward Secrecy; SFF3: Resilience to Key Compromise Impersonation; SFF4: Resilience to Replay Attacks; SFF5: Resilience to Modification Attacks; PEF1: Communication Cost; PEF2: Computation Cost; PEF3: Memory Cost

-:Missing evaluation/ Not applicable; ✓:secure against a particular attack or support a particular feature; ×: insecure or does not support a particular feature

* : Not standardized protocols for IoT

is required, in order to ascertain the impact of the proposal on the resources of constrained IoT devices. Finally, we must note that a few solutions do not detail the IoT protocol with which its particular approach could be integrated. This is relevant, particularly from the point of view of their applicability to practical IoT applications, given that an extensive and complete evaluation of a particular solution is only viable when considering the requirements of its integration with IoT protocols. Research can also explore how the proposed approaches may help in bringing light into how key bootstrapping using asymmetric cryptography may evolve in the context of the IoT security infrastructure. In the next section we extend our discussion on the research challenges and opportunities to approach key bootstrapping in the IoT.

VII. RESEARCH CHALLENGES AND APPROACHES FOR KEY BOOTSTRAPPING IN THE IoT

We proceed by discussing the main research challenges regarding the design of key boostrapping solutions, applicable to the requirements of IoT environments and the constraints in terms of resources that typically characterize sensing and actuating devices. We also highlight some of

the promising new approaches and future research directions that are being pursued in the area. As we discuss next, new approaches and proposals may involve the usage of optimized asymmetric protocols, blockchain technology, hardware-based and post-quantum cryptography for key bootstrapping and exchange:

- *Optimizing asymmetric cryptography protocols:* Asymmetric protocols based on raw public keys are clearly not scalable, as we previously observes, and thus must be backed up with strong authentication mechanisms, in order to ensure the secure exchange of keys. The more scalable certificate-based methods also need to be optimized, primarily due to the fact that, with IEEE 128-byte 802.15.4 packets, PKI certificates are clearly too large for being used with IoT applications. In fact, the larger packet size increases the memory and communication overhead, and may also lead to unnecessary fragmentation and re-assembly of packets. For this reason, there is a need to reduce the PKI certificate size. Preliminary works in this context involve the compression of 6LoWPAN for DTLS, as in [61] and [63], as well as the compression of IKE headers, as in [25]. As discussed in [60] and [92], another recent approach involves the compression of X.509 certificates by considering only the required fields, while simultaneously using compression, via e.g. CBOR encoding. As for future research in this context, researchers may explore additional mechanisms to make the compressed certificates fully compatible with the existing Internet certification infrastructure and connected devices, followed by the experimental validation of new proposals to verify the effectiveness of the proposed solutions. We may also expect that less explored schemes, for example self-certified and certificateless schemes, need also to be implemented and evaluate in the context of IoT applications, in order to fully evaluate its adequateness to IoT constrained devices and communication environments. An extensive security analysis and the identification of the various application domains needs also to be considered in the context of the IoT. Due to its exceptional advantages over certificate and identity based schemes, self-certified and certificateless schemes clearly represent candidate solutions for the IoT.
- *Blockchain Technology and Key Exchange:* There are various lines of research currently offering promising approaches to provide secure key bootstrapping on the IoT, and one of note is the employment of Blockchain technology [105]. Blockchain is essentially a distributed ledger technology, where blocks of data are added based on a consensus protocol known as the proof-of-work. Each block contains the hash of the previous block and, therefore, is linked to the previous block, giving the blockchain its name. Blockchain ensures the immutability or the integrity of the records stored in it, and is the main technology behind cryptocurrencies such as Bitcoin and Ethereum, and both these blockchains are decentralized and public, which comes at the cost of confidentiality. The usage of public blockchains may lead to privacy issues, unless a private blockchain is implemented, allowing the addition of records from authorized participants only.
  In the context of the IoT, the decentralized nature of blockchain provides a promising solution to ensure security in the IoT. Sedrati *et al.* [106] investigate blockchain solutions suitable for the IoT, including IoTA, KSI Guardtime, IBM Private Blockchain and ENIGMA. Whilst some of them ensure data integrity, others ensure confidentiality. However, it must be noted that the application of blockchain to the IoT is still in inception stage, and there is a long path before we can take full benefit of blockchain in resource-constrained IoT environments. In general, and if resources permit, blockchain may prove to be a suitable choice to ensure trust among IoT devices, and key bootstrapping in the context of blockchain-enabled security solutions is certainly an area presenting research challenges.
- *Hardware Based Solutions:* Due to their inherent benefits, hardware-assisted cryptography will continue to play an important role in the enabling of security in constrained IoT sensing and actuating devices. In what respects the employment of symmetric cryptography, this is already visible in the support of AES/CCM hardware-assisted encryption to support IEEE 802.15.4 and ZigBee, in various constrained sensing platforms [34]. In this context, we may expect hardware-assisted cryptography to also play a significant role in supporting future key management and bootstrapping solutions based on asymmetric cryptography.
  In the literature, Kothmayr *et al.* [13] implemented DTLS with RSA on sensing platforms enabled with a Trusted Platform Module (TPM). A TMP is an embedded chip capable of securely storing cryptographic keys and performing cryptographic operations efficiently. Even though it improves efficiency and performance, deploying hardware-accelerated cryptography on sensing devices may be complex and expensive and, thus, more research regarding the viability of hardware accelerators and its security analysis is required. Another popular alternative is to use Physical Unclonable Functions (PUF) to support key exchange operations. PUF-assisted operations utilize the inherent randomness that appears in the manufacturing process of the device and, thus, may be of assistance in providing unique identities to millions of devices supporting future IoT applications. The development and usage of PUF to support lightweight authentication may contribute further to support security in upcoming IoT implementations, and we observe that PUF-based authentication for the IoT is discussed in [107]–[109], and the integration of the unique identities provided by PUF with the key exchange protocol is a promising domain which deserves further exploration efforts.

- *Post-Quantum Cryptography and Key Exchange:* In this survey we have analyzed and reviewed public key cryptography solutions for key exchange in the IoT and, as we have previously discussed, the security of existing PKC solutions lie in the difficulty of solving hard mathematical problems with existing computers. However, the advancement in the development of quantum computers promises to threat this status quo. In this context, another type of cryptography comes into picture, currently referred to as post-quantum cryptography (PQC), which is secure against attacks by a quantum computer. PQC does not entail any special hardware and operates similarly to classic cryptography, although being based on mathematical problems which are infeasible, even for large quantum computers. Various families of post-quantum cryptography include hash-based, code-based, multi-variate polynomial and lattice-based cryptography [110]. Quantum computers with enough qubits, which are able to solve the mathematical problems that are at the foundation of PKC, have not been built yet, but research needs to prepare existing systems to be resistant against quantum attacks. This will involve significant efforts in ensuring a smooth migration from PKC to PQC-based security, since the former has supported already more than two decades of system development and deployments. In general, a secure IoT framework needs to be developed, in which all the existing algorithms are made quantum-safe, hence making the IoT crypto resilient, and this is certainly a new and exciting research area worth pursuing. In this context, the research on PQC has gained much attention from the industry, government and academia recently. Recent announcements by the NIST and US National Security Agency (NSA) identify the need to consider the transition to PQC schemes [111], and some researchers have also focused on quantum cryptography for the IoT. For example, Cheng *et al.* [112] discuss the importance of securing IoT in the quantum world, and ongoing projects, such as PQCRYPTO EU-Project, currently research the applicability of PQC to IoT devices and applications [113]. Besides, Crypto-MathCREST [114], a research project supported by Japan Science and Technology Agency, focuses on the study of mathematical problems underlying the security of PQC. Whilst ongoing efforts to develop PQC solutions for IoT are clear, more work related to the design and evaluation of PQC for the IoT is required, and this certainly includes the design of new solutions to support key negotiation and boostrapping in IoT devices.

VIII. CONCLUSION

In this survey we focus on the important aspect of secure key bootstrapping, particularly in what respects its usage in the context of IoT applications employing constrained sensing and actuating devices. We perform an analysis of the existing IoT security protocols and technologies at the different layers of the IoT communication stack focusing, in particular, at how key bootstrapping may be effectively guaranteed to take place with security, in the context of the various communications and security solutions. Given its significance in terms of scalability to support future IoT applications, we focus our discussion particularly in asymmetric or public key-based key bootstrapping solutions. Our analysis is structured around a taxonomy of the existing classification approaches and protocols, and the various research proposals are analyzed along different core functionality and security features.

As previously noted, previous works not targeting IoT protocols can be revisited, evaluated and even evolve towards its integration with the Internet communications infrastructure. We also identify and discuss further research approaches and opportunities, particularly in what respects exploring the optimization of asymmetric cryptographic protocols, or the usage of blockchain, hardware-based and post-quantum cryptography to support key bootstrapping. As we have noted, there is currently is a lack of a survey focused on this important class of key bootstrapping solutions for the IoT and, other than the analysis of the existing research proposals, we identify and discuss opportunities to conduct further research in this important area of IoT security.

REFERENCES

[1] C. M. VNI, "Cisco visual networking index: Global mobile data traffic forecast update, 2016–2021," White Paper, San Jose, CA, USA, 2017.

[2] C. Bormann, M. Ersue, and A. Keranen, "Terminology for constrained-node networks," IETF, Fremont, CA, USA, Tech. Rep. RFC 7228, May 2014.

[3] J. A. Stankovic, "Research directions for the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 3–9, Feb. 2014.

[4] J. Granjal, E. Monteiro, and J. S. Silva, "Security for the Internet of Things: A survey of existing protocols and open research issues," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 3, pp. 1294–1312, Aug. 2015.

[5] D. Verma, R. Jain, and A. Shrivastava, "Performance analysis of cryptographic algorithms rsa and ecc in wireless sensor networks," *IUP J. Telecommun.*, vol. 7, no. 3, p. 51, 2015.

[6] O. Garcia-Morchon, S. Kumar, and M. Sethi, *State-of-the-Art and Challenges for the Internet of Things Security (Work in Progress)*, Internet-Draft, draft-IRTF-t2trg-IoT-seccons-11, IETF Secretariat, Feb. 2017.

[7] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Tech. J.*, vol. 28, no. 4, pp. 656–715, Oct. 1949.

[8] A. He and B. Sarikaya, *IoT Security Bootstrapping: Survey and Design Considerations (Work in Progress)*, Internet-Draft, draft-he-6lo-analysis-IoT-sbootstrapping-00, IETF Secretariat, Mar. 2015.

[9] B. Sarikaya, Y. Ohba, R. Moskowitz, Z. Cao, and R. Cragie, *Security Bootstrapping Solution for Resource-Constrained Devices (Work in Progress)*, Internet-Draft, draft-sarikaya-core-sbootstrapping-05, IETF Secretariat, Jul. 2012.

[10] U. M. Maurer and S. Wolf, "The diffie–hellman protocol," *Des., Codes Cryptogr.*, vol. 19, nos. 2–3, pp. 147–171, 2000.

[11] Q. Zhang and Z. Shi, "A new way to prevent uks attacks using hardware security chips," *Int. J. Netw. Secur.*, vol. 19, no. 5, pp. 823–831, 2017.

[12] N. Jansma and B. Arrendondo, *Performance Comparison of Elliptic Curve and RSA Digital Signatures*, NICJ. Net/Files, 2004.

[13] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based security and two-way authentication for the Internet of Things," *Ad Hoc Netw.*, vol. 11, no. 8, pp. 2710–2723, 2013.

[14] P. Levis *et al.*, "TinyOS: An operating system for sensor networks," in *Ambient Intelligence*. Berlin, Germany: Springer, 2005, pp. 115–148.

[15] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman, *Mikey: Multimedia Internet keying*, document RFC 3830, IETF, Fremont, CA, USA, 2004.

[16] K. T. Nguyen, N. Oualha, and M. Laurent, "Novel lightweight signcryption-based key distribution mechanisms for MIKEY," in *Information Security Theory and Practice*, S. Foresti and J. Lopez, Eds. Cham, Switzerland: Springer, 2016, pp. 19–34.
[17] R. Moskowitz, T. Heer, P. Jokela, and T. Henderson, *Host Identity Protocol Version 2 (HIPV2)*, document RFC 7401, IETF, Fremont, CA, USA, 2015.
[18] A. Gupta and E. R. K. Jha, "A survey of 5G network: Architecture and emerging technologies," *IEEE Access*, vol. 3, pp. 1206–1232, Jul. 2015.
[19] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 packets over IEEE 802.15. 4 networks," IETF, Fremont, CA, USA, Tech. Rep. RFC 4944, Sep. 2007.
[20] M. R. Palattella *et al.*, "Standardized protocol stack for the Internet of (Important) Things," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 3, pp. 1389–1406, 3rd Quart., 2013.
[21] J. A. Gutierrez, M. Naeve, E. Callaway, M. Bourgeois, V. Mitter, and B. Heile, "IEEE 802.15. 4: A developing standard for low-power low-cost wireless personal area networks," *IEEE Netw.*, vol. 15, no. 5, pp. 12–19, Sep. 2001.
[22] A. Zigbee, *Zigbee Specification*, document 053474r13, ZigBee, 2006.
[23] A. N. Kim, F. Hekland, S. Petersen, and P. Doyle, "When hart goes wireless: Understanding and implementing the wirelesshart standard," in *Proc. IEEE Int. Conf. Emerg. Technol. Factory Automat.*, Sep. 2008, pp. 899–907.
[24] J. Granjal, R. Silva, E. Monteiro, J. S. Silva, and F. Boavida, "Why is ipsec a viable option for wireless sensor networks," in *Proc. 5th IEEE Int. Conf. Mobile Ad Hoc Sensor Syst.*, Sep./Oct. 2008, pp. 802–807.
[25] S. Raza, T. Voigt, and V. Jutvik, "Lightweight IKEv2: A key management solution for both the compressed IPsec and the IEEE 802.15. 4 security," in *Proc. IETF Workshop Smart Object Secur.*, vol. 23, 2012, pp. 1–2.
[26] J. Nieminen, T. Savolainen, M. Isomaki, B. Patil, Z. Shelby, and C. Gomez, *IPv6 Over Bluetooth (R) Low Energy*, document RFC 7668, IETF, Fremont, CA, USA, 2015.
[27] T. Winter *et al.*, *Rpl: Ipv6 Routing Protocol for Low-Power and Lossy Networks*, document RFC 6550, IETF, Fremont, CA, USA, Mar. 2012.
[28] J. Granjal, E. Monteiro, and J. S. Silva, "Security in the integration of low-power Wireless Sensor Networks with the Internet: A survey," *Ad Hoc Netw.*, vol. 24, pp. 264–287, Jan. 2015.
[29] Z. Shelby, K. Hartke, and C. Bormann, *IETF RFC 7252*, The Constrained Application Protocol (CoAP), 2014.
[30] E. Rescorla and N. Modadugu, *RFC 6347: Datagram Transport Layer Security Version 1.2*, Internet Eng. Task Force, Fremont, CA, USA, 2012.
[31] G. Selander, J. Mattsson, F. Palombini, and L. Seitz, *Object Security of COAP (Oscoap) (Work in Progress)*, Internet-Draft draft-IETF-core-object-security-04, IETF Secretariat, Jul. 2017.
[32] J. Granjal, E. Monteiro, and J. S. Silva, "Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications," in *Proc. Int. Conf. Wired/Wireless Internet Commun.* St. Petersburg, Russia: Springer, 2013, pp. 140–153.
[33] M. A. Simplício, Jr., P. S. L. M. Barreto, C. B. Margi, and T. C. M. B. Carvalho, "A survey on key management mechanisms for distributed wireless sensor networks," *Comput. Netw.*, vol. 54, no. 15, pp. 2591–2612, 2010.
[34] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "Key management systems for sensor networks in the context of the Internet of Things," *Comput. Elect. Eng.*, vol. 37, no. 2, pp. 147–159, 2011.
[35] Y. B. Saied, "Collaborative security for the Internet of Things," Ph.D. dissertation, Dept. Réseaux et Services Multimédia Mobiles, Institut Nat. des Télécommunications, Évry, France, 2013.
[36] K. T. Nguyen, M. Laurent, and N. Oualha, "Survey on secure communication protocols for the Internet of things," *Ad Hoc Netw.*, vol. 32, pp. 17–31, Sep. 2015.
[37] B. Sarikaya, M. Sethi, and D. Garcia-Carillo, *Secure IoT Bootstrapping: A Survey (Work in Progress)*, Internet-Draft draft-sarikaya-t2trg-sbootstrapping-05, IETF Secretariat, Sep. 2018.
[38] A. K. Das, S. Zeadally, and D. He, "Taxonomy and analysis of security protocols for Internet of Things," *Future Gener. Comput. Syst.*, vol. 89, pp. 110–125, Dec. 2018.
[39] A. Menezes, J. Katz, P. van Oorschot, and S. Vanstone, *Handbook of Applied Cryptography*. Boca Raton, FL, USA: CRC Press, 1997.
[40] B. C. Neuman and T. Ts'o, "Kerberos: An authentication service for computer networks," *IEEE Commun. Mag.*, vol. 32, no. 9, pp. 33–38, Sep. 1994.
[41] R. M. Needham and M. D. Schroeder, "Using encryption for authentication in large networks of computers," *Commun. ACM*, vol. 21, no. 12, pp. 993–999, Dec. 1978.
[42] C. Boyd and A. Mathuria, *Protocols for Authentication and Key Establishment*. Berlin, Germany: Springer, 2003.
[43] S. Blake-Wilson, D. Johnson, and A. Menezes, "Key agreement protocols and their security analysis," in *Proc. IMA Int. Conf. Cryptogr. Coding*. Cirencester, U.K.: Springer, 1997, pp. 30–45.
[44] A. Menezes, "Some new key agreement protocols providing implicit authentication," in *Proc. Workshop Sel. Areas Cryptogr.* Boca Raton, FL, USA: CRC Press, 1997, pp. 22–32.
[45] Y. Zheng, "Digital signcryption or how to achieve cost (signature & encryption)≪ cost (signature)+ cost (encryption)," in *Proc. Annu. Int. Cryptol. Conf.* Springer, 1997, pp. 165–179.
[46] F. Li, Z. Zheng, and C. Jin, "Secure and efficient data transmission in the Internet of Things," *Telecommun. Syst.*, vol. 62, no. 1, pp. 111–122, 2016.
[47] A. Karati, S. H. Islam, G. Biswas, M. Z. A. Bhuiyan, P. Vijayakumar, and M. Karuppiah, "Provably secure identity-based signcryption scheme for crowdsourced industrial Internet of Things environments," *IEEE Internet Things J.*, vol. 5, no. 4, pp. 2904–2914, Aug. 2018.
[48] M. Turkanović, B. Brumen, and M. Hölbl, "A novel user authentication and key agreement scheme for heterogeneous ad hoc wireless sensor networks, based on the Internet of Things notion," *Ad Hoc Netw.*, vol. 20, pp. 96–112, Sep. 2014.
[49] A. Menezes and B. Ustaoglu, "On the importance of public-key validation in the MQV and HMQV key agreement protocols," in *Proc. Int. Conf. Cryptol.* Kolkata, India: Springer, 2006, pp. 133–147.
[50] M. Campagna, "Standards for efficient cryptography sec 4: Elliptic curve QU-Vanstone implicit certificate Scheme (ECQV)," Certicom, Mississauga, ON, Canada, Tech. Rep., Jan. 2013.
[51] D. Chen, G. Chang, D. Sun, J. Jia, and X. Wang, "Lightweight key management scheme to enhance the security of Internet of Things," *Int. J. Wire. Mob. Comput.*, vol. 5, pp. 191–198, May 2012.
[52] M. A. M. Isa, N. N. Mohamed, H. Hashim, S. F. S. Adnan, J. A. Manan, and R. Mahmod, "A lightweight and secure TFTP protocol for smart environment," in *Proc. Int. Symp. Comput. Appl. Ind. Electron. (ISCAIE)*, Dec. 2012, pp. 302–306.
[53] R. Hummen, H. Wirtz, J. H. Ziegeldorf, J. Hiller, and K. Wehrle, "Tailoring end-to-end IP security protocols to the Internet of Things," in *Proc. IEEE ICNP*, Oct. 2013, pp. 1–10.
[54] R. Hummen, J. Hiller, M. Henze, and K. Wehrle, "Slimfit—A HIP DEX compression layer for the IP-based Internet of things," in *Proc. IEEE 9th Int. Conf. Wireless Mobile Comput., Netw. Commun. (WiMob)*, Lyon, France, Oct. 2013, pp. 259–266.
[55] S. Sahraoui and A. Bilami, "Efficient hip-based approach to ensure lightweight end-to-end security in the Internet of Things," *Comput. Netw.*, vol. 91, pp. 26–45, Nov. 2015.
[56] F. V. Meca, J. H. Ziegeldorf, P. M. Sanchez, O. G. Morchon, S. S. Kumar, and S. L. Keoh, "HIP security architecture for the IP-based Internet of Things," in *Proc. 27th Int. Conf. Adv. Inf. Netw. Appl. Workshops*, Barcelona, Spain, Mar. 2013, pp. 1331–1336.
[57] S. Kalra and S. K. Sood, "Secure authentication scheme for IoT and cloud servers," *Pervas. Mobile Computing*, vol. 24, pp. 210–223, Dec. 2015.
[58] M. Schukat and P. Cortijo, "Public key infrastructures and digital certificates for the Internet of Things," in *Proc. 26th Irish Signals Syst. Conf. (ISSC)*, Jun. 2015, pp. 1–5.
[59] S. Misra, S. Goswami, C. Taneja, A. Mukherjee, and M. S. Obaidat, "A PKI adapted model for secure information dissemination in industrial control and automation 6LoWPANs," *IEEE Access*, vol. 3, pp. 875–889, 2015.
[60] F. Forsby, M. Furuhed, P. Papadimitratos, and S. Raza, "Lightweight X. 509 digital certificates for the Internet of Things," in *Interoperability, Safety and Security in IoT*. Springer, 2017, pp. 123–133.
[61] S. Raza, D. Trabalza, and T. Voigt, "6LoWPAN compressed DTLS for CoAP," in *Proc. IEEE 8th Int. Conf. Distrib. Comput. Sensor Syst.*, May 2012, pp. 287–289.
[62] R. Hummen, J. H. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards viable certificate-based authentication for the Internet of Things," in *Proc. 2nd ACM Workshop Hot Topics Wireless Netw. Secur. Privacy (HotWiSec)*, Budapest, Hungary: ACM, 2013, pp. 37–42.
[63] S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight secure CoAP for the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3711–3720, Oct. 2013.

[64] M. Kasraoui, A. Cabani, and H. Chafouk, "Collaborative key exchange system based on chinese remainder theorem in heterogeneous wireless sensor networks," *Int. J. Distrib. Sensor Netw.*, vol. 11, no. 11, 2015, Art. no. 159518.

[65] P. Porambage, P. Kumar, A. Gurtov, M. Ylianttila, and E. Harjula, *Certificate Based Keying Scheme for DTLS Secured IoT (Work in Progress)*," Internet-Draft Draft-PPORAMBA-DTLS-Certkey-00, IETF Secretariat, Jun. 2013.

[66] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "Two-phase authentication protocol for wireless sensor networks in distributed IoT applications," in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Apr. 2014, pp. 2728–2733.

[67] P. Porambage, C. Schmitt, P. Kumar, A. Gurtov, and M. Ylianttila, "PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 7, 2014, Art. no. 357430.

[68] S. Sciancalepore, A. Capossele, G. Piro, G. Boggia, and G. Bianchi, "Key management protocol with implicit certificates for IoT systems," in *Proc. Workshop IoT Challenges Mobile Ind. Syst.* New York, NY, USA: ACM, 2015, pp. 37–42.

[69] D. A. Ha, K. T. Nguyen, and J. K. Zao, "Efficient authentication of resource-constrained IoT devices based on ECQV implicit certificates and datagram transport layer security protocol," in *Proc. 7th Symp. Inf. Commun. Technol.* New York, NY, USA: ACM, 2016, pp. 173–179.

[70] S. Sciancalepore, G. Piro, G. Boggia, and G. Bianchi, "Public key authentication and key agreement in IoT devices with minimal airtime consumption," *IEEE Embedded Syst. Lett.*, vol. 9, no. 1, pp. 1–4, Mar. 2017.

[71] M. A. Simplicio, Jr., M. V. M. Silva, R. C. A. Alves, and T. K. C. Shibata, "Lightweight and escrow-less authenticated key agreement for the Internet of Things," *Comput. Commun.*, vol. 98, pp. 43–51, Jan. 2017.

[72] C.-S. Park, "A secure and efficient ECQV implicit certificate issuance protocol for the Internet of Things applications," *IEEE Sensors J.*, vol. 17, no. 7, pp. 2215–2223, Apr. 2017.

[73] H. K.-H. So, S. H. M. Kwok, E. Y. Lam, and K.-S. Lui, "Zero-configuration identity-based signcryption scheme for smart grid," in *Proc. 1st IEEE Int. Conf. Smart Grid Commun.*, Oct. 2010, pp. 321–326.

[74] W. Chen, "An IBE-based security scheme on Internet of Things," in *Proc. IEEE 2nd Int. Conf. Cloud Comput. Intell. Syst.*, vol. 3, Oct./Nov. 2012, pp. 1046–1049.

[75] F. Li and P. Xiong, "Practical secure communication for integrating wireless sensor networks into the Internet of Things," *IEEE Sensors J.*, vol. 13, no. 10, pp. 3677–3684, Oct. 2013.

[76] V. L. Shivraj, M. A. Rajan, M. Singh, and P. Balamuralidhar, "One time password authentication scheme based on elliptic curves for Internet of Things (IoT)," in *Proc. 5th Nat. Symp. Inf. Technol., Towards New Smart World (NSITNSW)*, Riyadh, Saudi Arabia, Feb. 2015, pp. 1–6.

[77] J.-X. Hu, C.-L. Chen, C.-L. Fan, and K.-H. Wang, "An intelligent and secure health monitoring scheme using IoT sensor based on cloud computing," *J. Sensors*, vol. 2017, Jan. 2017, Art. no. 3734764.

[78] Q. Jiang, S. Zeadally, J. Ma, and D. He, "Lightweight three-factor authentication and key agreement protocol for Internet-integrated wireless sensor networks," *IEEE Access*, vol. 5, pp. 3376–3392, 2017.

[79] M. E. S. Saeed, Q.-Y. Liu, G. Tian, B. Gao, and F. Li, "AKAIoTs: Authenticated key agreement for Internet of Things," in *Wireless Networks*. Mar. 2018.

[80] A. P. Haripriya and K. Kulothungan, "ECC based self-certified key management scheme for mutual authentication in Internet of Things," in *Proc. Int. Conf. Emerg. Technol. Trends (ICETT)*, Kollam, India, Oct. 2016, pp. 1–6.

[81] S. H. Seo, J. Won, S. Sultana, and E. Bertino, "Effective key management in dynamic wireless sensor networks," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 2, pp. 371–383, Feb. 2015.

[82] P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, and T. Kivinen, "Using raw public keys in transport layer security (TLS) and datagram transport layer security (DTLS)," IETF, Fremont, CA, USA, Tech. Rep. RFC 7250, Jun. 2014.

[83] M. O. Rabin, "Digitalized signatures and public-key functions as intractable as factorization," Massachusetts Inst of Tech Cambridge Lab for Computer Science, Cambridge, MA, USA, Tech. Rep., 1979.

[84] S. Mirzadeh, H. Cruickshank, and R. Tafazolli, "Secure device pairing: A survey," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 17–40, 1st Quart., 2014.

[85] P. Hoffman and J. Schlyter, *The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA*, document RFC 6698, RFC Editor, Aug. 2012.

[86] R. Alexander and T. Tsao, *Adapted Multimedia Internet Keying (AMIKEY): An Extension of Multimedia Internet Keying (MIKEY) Methods for Generic LLN Environments (Work in Progress)*, Internet-Draft draft-alexander-roll-mikey-lln-key-mgmt-04, IETF Secretariat, Sep. 2012.

[87] C. Schmitt, T. Kothmayr, B. Ertl, W. Hu, L. Braun, and G. Carle, "TinyIP-FIX: An efficient application protocol for data exchange in cyber physical systems," *Comput. Commun.*, vol. 74, pp. 63–76, Jan. 2016.

[88] J. A. Buchmann, E. Karatsiolis, and A. Wiesmaier, *Introduction to Public Key Infrastructures*. New York, NY, USA: Springer-Verlag, 2013.

[89] D. McGrew and M. Pritikin, *The Compressed X.509 Certificate Format (Work in Progress)*, Internet-Draft draft-pritikin-comp-X509-00, IETF Secretariat, May 2010.

[90] L. P. Deutsch, "Deflate compressed data format specification version 1.3," Tech. Rep. RFC 1951, RFC Editor, May 1996.

[91] G. Edgecombe. (Dec. 2016). *Compressing X.509 Certificates*. Accessed: Sep. 30, 2018. [Online]. Available: https://www.grahamedgecombe.com/blog/2016/12/22/compressing-x509-certificates

[92] H. Kwon, S. Raza, and J. G. Ko, "Poster: On compressing PKI certificates for resource limited Internet of Things devices," in *Proc. Asia Conf. Comput. Commun. Secur. (ASIACCS)*. New York, NY, USA: ACM, 2018, pp. 837–839.

[93] Certicom, "Explaining implicit certificates," Certicom, Mississauga, ON, Canada, Tech. Rep., 2016.

[94] X. Vilajosana, P. Tuset, T. Watteyne, and K. Pister, "OpenMote: Open-source prototyping platform for the industrial IoT," in *Ad Hoc Network*, Cham, Switzerland: Springer, 2015, pp. 211–222.

[95] A. P. Sarr, P. Elbaz-Vincent, and J.-C. Bajard, "A new security model for authenticated key agreement," in *Proc. Int. Conf. Secur. Cryptogr. Netw.* Amalfi, Italy: Springer, 2010, pp. 219–234.

[96] A. Shamir and D. Chaum, "Identity-based cryptosystems and signature schemes," in *Advances in Cryptology*. Berlin, Germany: Springer, 1984, pp. 47–53.

[97] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Advances in Cryptology-CRYPTO*, J. Kilian, Ed. Berlin, Germany: Springer, 2001, pp. 213–229.

[98] B. S. Adiga, P. Balamuralidhar, M. A. Rajan, R. Shastry, and V. L. Shivraj, "An identity based encryption using elliptic curve cryptography for secure m2m communication," in *Proc. 1st Int. Conf. Secur. Internet Things*. New York, NY, USA: ACM, 2012, pp. 68–74.

[99] A. L. John and S. M. Thampi, "Encryption scheme based on hyperelliptic curve cryptography," in *Proc. Int. Conf. Secur., Privacy Anonymity Comput., Commun. Storage*. Zhangjiajie, China: Springer, 2016, pp. 491–506.

[100] M. Girault, "Self-certified public keys," in *Proc. Workshop Theory Appl. Cryptograph. Techn.* Brighton, U.K.: Springer, 1991, pp. 490–497.

[101] S. Saeednia, "Identity-based and self-certified key-exchange protocols," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Sydney, NSW, Australia: Springer, 1997, pp. 303–313.

[102] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Advances in Cryptology-ASIACRYPT*, C.-S. Laih, Ed. Berlin, Germany: Springer, 2003, pp. 452–473.

[103] K.-H. Yeh, C. Su, K.-K. R. Choo, and W. Chiu, "A novel certificateless signature scheme for smart objects in the Internet-of-Things," *Sensors*, vol. 17, no. 5, p. 1001, 2017.

[104] M. Ma, D. He, N. Kumar, K.-K. R. Choo, and J. Chen, "Certificateless searchable public key encryption scheme for industrial Internet of Things," *IEEE Trans. Ind. Informat.*, vol. 14, no. 2, pp. 759–767, Feb. 2018.

[105] A. Dorri, S. S. Kanhere, and R. Jurdak. (2016). "Blockchain in Internet of Things: Challenges and solutions." [Online]. Available: https://arxiv.org/abs/1608.05187

[106] A. Sedrati, M. A. Abdelraheem, and S. Raza, "Blockchain and IoT: Mind the Gap," in *Interoperability, Safety and Security in IoT*. Cham, Switzerland: Springer, 2017, pp. 113–122.

[107] T. Idriss, H. Idriss, and M. Bayoumi, "A PUF-based paradigm for IoT security," in *Proc. IEEE 3rd World Forum Internet Things (WF-IoT)*, Dec. 2016, pp. 700–705.

[108] U. Chatterjee *et al.*, "Building PUF based authentication and key exchange protocol for IOT without explicit CRPs in verifier database," *IEEE Trans. Dependable Secure Comput.*, vol. 10, no. 8, 2018.

[109] A. Braeken, "PUF based authentication protocol for IoT," *Symmetry*, vol. 10, no. 8, p. 352, 2018.
[110] J. Buchmann, K. Lauter, and M. Mosca, "Postquantum cryptography-State of the art," *IEEE Security Privacy*, vol. 15, no. 4, pp. 12–13, 2017.
[111] L. Chen *et al.*, *Rep. post-quantum cryptography*. Gaithersburg, MD, USA: Nat. Inst. Standards Technol., 2016.
[112] C. Cheng, R. Lu, A. Petzoldt, and T. Takagi, "Securing the Internet of Things in a quantum world," *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 116–120, Feb. 2017.
[113] (2018). *PQCRYPTO ICT-645622 Research Project*. [Online]. Available: https://pqcrypto.eu.org
[114] *Cryptomathcrest JST Research Project*. Accessed: 2018. [Online]. Available: http://crypto.mist.i.u-tokyo.ac.jp/crest/english

MANISHA MALIK was born in India, in 1990. She received the B.E. degree in computer science and engineering from Chitkara University, in 2012, and the M.E. degree in computer science and engineering from Panjab University, in 2015. She is currently pursuing the Ph.D. degree in computer science and engineering with the National Institute of Technical Teachers' Training and Research, Chandigarh, India (Panjab University's affiliated institute).

During her post-graduation, she developed keen interest in the area of cyber security and forensics and has delivered a number of lectures for the teachers of polytechnic and engineering colleges. She is currently working on the security of the Internet of Things (IoT). Her research interests include communication and network security in the IoT, and the design and implementation of lightweight cryptographic primitives for the next-generation IoT networks.

MAITREYEE DUTTA was born in Guwahati, India. She received the B.E. degree in electronics and communication engineering from Assam Science and Technology University, and the M.E. degree in electronics and communication engineering and the Ph.D. degree with specialization in image processing from Panjab University.

She is currently a Professor with the Computer Science and Engineering Department, National Institute of Technical Teachers' Training and Research, Chandigarh, India. She has over 18 years of teaching experience. Her research interests include digital signal processing, advanced computer architecture, data warehousing and mining, image processing, and the Internet of Things. She has more than 100 research publications in reputed journals and conferences. She completed one sponsored research project—Establishment of Cyber Security Lab—funded by the Ministry of IT, Government of India, New Delhi, amounting Rs. 45.65 lac.

JORGE GRANJAL (S'10–M'12) received the Ph.D. degree, in 2014. He is currently an Assistant Professor with the Department of Informatics Engineering, Faculty of Science and Technology, University of Coimbra, Portugal, where he is also a Researcher of the Laboratory of Communication and Telematics, Centre for Informatics and Systems. His main current research interests include computer networks, network security, and wireless sensor networks. Jorge is also a member of ACM communications groups.