Published in: IEEE Transactions on Industrial Informatics journal.

The Version of Record is available online at: https://doi.org/10.1109/TII.2018.2791956

Integrity for an Event Notification Within the Industrial Internet of Things by Using Group Signatures

Christian Esposito, *Member, IEEE*, Aniello Castiglione, *Member, IEEE*, Francesco Palmieri, and Alfredo De Santis

***Abstract*—In the last years, several academic research efforts have focused on security requirements, threat models, and attack taxonomies concerning the application of the Internet of Things (IoT) in critical systems. Since such systems are strongly data intensive, it is of pivotal importance to provide integrity for the messages moving throughout the IoT infrastructure by means of publish/subscribe services. Integrity provisioning in industrial IoT scenarios has received marginal attention with respect to other primary security features. The existing solutions are lacking the needed focus on the peculiarities of the event notification and on the demand introduced by resource-constrained devices. This work contributes by applying group signatures so as to avoid managing certificates, violating the spatial decoupling, or implying an excessive resource usage. A proof-of-concept prototype of the proposed solution has been realized for platforms based on TinyOS, and simulations with TOSSIM have been conducted in order to empirically assess its performance and effectiveness.**

***Index Terms*—Group signature, identity-based cryptosystems, message integrity, publish/subscribe service.**

I. Introduction

INTERNET of Things (IoT) [1] can be simply described as the integration of wireless sensor networks with cloud computing, where smart sensing nodes (or actuators) located on the network edge monitor (or control) the physical environment by eventually performing some initial preprocessing on the gathered data. Such data move from the edge toward the network core by reaching the cloud in order to be persistently stored and analyzed by generating new information or taking proper decisions to control a given process. Differently from the devices operating in traditional sensor network scenarios, many IoT nodes are equipped with long- or short-range wireless communication interfaces, with IP capabilities, in order to connect to a base station, a local router, or an access point (often referred as "gateway"), providing Internet access and, hence, allowing the node to directly reach the cloud. This removes the need of a sink node collecting the sensory data and forwarding them to Internet-accessible remote processing services. However, it is possible to have IoT nodes not directly connected to the cloud, but having some intermediaries along the way, performing some sort of preprocessing, filtering, or aggregation and, hence, leading to the so-called fog computing architectures [2]. This means pushing the frontier of processing applications and analytics away from centralized nodes by distributing processing intelligence near to the true origins of the data of interest [3].

The authors are with the Department of Computer Science, University of Salerno, 84084 Fisciano, Italy (e-mail: christian.esposito@dia.unisa.it; castiglione@ieee.org; fpalmieri@unisa.it; ads@unisa.it).

A. Publish/Subscribe Services for the Industrial IoT

Such a multitude of nodes within an IoT infrastructure are characterized by a different communication pattern complementing the more traditional request/reply mechanism implemented by web services needed for their direct referencing by users and/or applications. For scalability and seamless mobility reasons, it is important to avoid the necessity of static or rigidly established interconnections among the IoT nodes and intermediaries, so as to apply a plug-and-play approach for the automatic detection of a novel node and the establishment of a new connection. For this reason, the publish/subscribe paradigm [4] has imposed itself as the best communication scheme to convey data within an IoT system, thanks to its decoupling, asynchrony, and flexibility features. Moreover, such a scheme natively supports data-centric communications, rather than network-centric ones, which perfectly matches the event-driven model of the sensors where nodes express their interest by means of subscriptions that are string-matching predicates on the notification contents or their topics. Within the current panorama of communication middleware for IoT scenarios, there are several solutions providing an implementation of the publish/subscribe paradigm, as surveyed in [5]. Most of them are based on well-formalized standards from the IETF, OMG, or OASIS and assume an infrastructure-based architecture with the presence of special nodes (i.e., characterized by more computational and storage resources than the IoT nodes), where notification brokers are hosted, to mediate among the IoT nodes by managing subscriptions and routing notifications to the interested subscribers. Also, infrastructureless solutions are present, despite

being mainly research prototypes, such as in [6], where nodes assume a promiscuous architecture without any brokers and where the notifications are managed in a decentralized way by the publishers and subscribers running on the IoT nodes. Such a second solution lies along the current research frontier, and it is more complex to implement, since it requires that the nodes have to self-organize themselves within a proper overlay organization but provides a higher degree of scalability, availability, and reliability due to the lack of brokers, which may represent a performance bottleneck and single point of failure for the overall infrastructure.

B. Need for Security in the Industrial IoT

The IoT is among the recent technologies that are paving the way for the fourth industrial revolution, named as Industry 4.0 [7], which, as the other revolutions, consists in a radical rethinking of the way manufacturing enterprises are being managed and/or the manufacturing processes are implemented within an enterprise [8]. The novelty of such a revolution is the pervasive role of information and communication technology (ICT) within the manufacturing in order to cope with the current requirements of higher productivity, lower costs, and the better planning of the overall process, even at a global scale. A concrete example of such an ICT-driven revolution in manufacturing is represented by the "Factories of the Future" [9], a public–private partnership under Horizon 2020 produced by European Factories of the Future Research Association. Such a partnership has produced a roadmap for introducing innovation-driven transformations within the European manufacturing sectors. Also, in the United States, a similar effort for the application of the emerging ICT technologies to the manufacturing sector has been established within the context of the Advanced Manufacturing Partnership (AMP) formed in 2011. The AMP has finalized a technical report [10], which contains recommendations for the innovations within the manufacturing domain to determine the most pressing challenges and transformation opportunities to improve the current manufacturing industries and to enhance higher global competitiveness. When IoT technologies are applied within the context of the manufacturing sectors to realize the concept of smart factories, as in [11], we refer to them as Industrial Internet of Things (IIoT) [12], and Fig. 1 schematically depicts a generic example of the IIoT within a manufactory production site. Specifically, the IIoT nodes are deployed within all the elements of a factory in order to monitor their behavior and can be coupled with proper actuator equipment in order to implement any recovery/control strategy determined by processing the monitoring data from the IIoT nodes together with a description of the running manufacturing process. As an example, let us consider a typical assembly line that brings semifinished products from one workstation to another, where the parts with proper transforming actions are added in sequence until the final product is built. A sensor is attached to each workstation for monitoring and commanding purposes, as illustrated in the figure, and interacts, directly or indirectly through some gateways, with the cloud-based sensor management facility.

The IIoT will have a crucial role within the smart factories [13], mainly supporting predictive maintenance and process

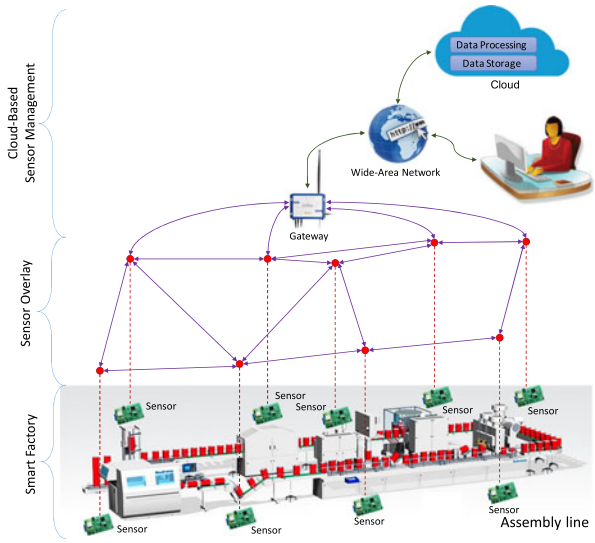
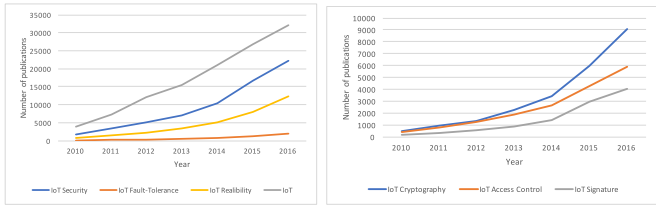


Fig. 1. Example of IIoT application.



(a) (b)

Fig. 2. Literature trend analysis within the last decade: (a) interest in the non-functional properties for IoT over the time and (b) interest in security-related aspects over the time.

optimization; therefore, it is important to have the IIoT able to tolerate faults and possible attacks so as to do not compromise the running manufacturing process by avoiding economic losses, bad publicity, environmental disasters, or causalities in human lives. This moves the current IIoT-related research focus from the integration of heterogeneous devices and technologies for scalable and energy-efficient data management/processing to the development of a solid framework providing reliability, availability, and security for the sensory data communication and for the involved hardware and software assets characterizing the IIoT architecture. In particular, the security of the IIoT is felt as a key research challenge [14], as resulted from an analysis of the current literature published within the last decades and retrievable via Google Scholar by using the keywords reported in the legend of the two charts in Fig. 2. Specifically, from Fig. 2(a), we can notice that security has a predominant trend over the other non-functional properties of fault tolerance and reliability. In fact, the current application of the IIoT to smart factories disruptily changes a reality that has been traditionally using closed networks, that for such reason, were considered secure from possible attacks. However, in order to tackle the global scale of several firms or the possibility of making coalitions among geographically sparse firms, multiple instances of smart factories have started to be interconnected by using the Internet, rising several novel and unseen security and vulnerability issues. Fig. 2(b) contains a further analysis of

the current research trends by focusing on the security-related aspects, where cryptography results to be predominant over access control and signature issues. This is due to the fact that the traditional cryptosystems are not easily deployable in IIoT scenarios due to resource limitations characterizing the involved devices. However, access control is also an active research topic, despite its long history and the number of models theorized and implemented over the years, since the IIoT is calling for novel access control strategies where multiple access control models must coexist, and more dynamic schemes are needed to counter masquerading attacks as well as the exposure of personal data or the traceability of the user habits. Achieving integrity for the exchanged data has been considered a minor concern and received lower attention, even if such a situation is changing in the last years. Indeed, multiple kinds of attacks [15], such as man-in-the-middle, false data injection, or message replay, potentially affect IIoT infrastructures and can be faced only if the integrity of the exchanged data is preserved and their source is strongly authenticated and traceable.

C. Integrity Enforcement and Open Issues

When the integrity of exchanged messages has to be verified, techniques based on a digital signature must be used. They generically consist in a publisher computing some kind of hash on the data to be exchanged and associating it to the outgoing notification after encrypting it with its private key. On the other hand, the subscriber can verify the integrity by computing the hash of the message, decrypting the hash contained in the received notification by using the public key of the publisher, and checking if these two hashes coincide. Typically, a public key infrastructure (PKI) is used so that the entity that needs to verify the received digital signature can obtain the public key of the signer, so that the identity of the signer is documented by a valid digital certificate [16]. Such a basic scheme underlying every specific solution for a digital signature implies several issues when applied within the context of publish/subscribe services, as investigated in [17]. Summarizing the reported findings, the current solutions for digital signatures in publish/subscribe services exhibit two main problems: on the one hand, we can find some overheads and inefficiencies in retrieving and keeping certificates; on the other hand, there are identity exposures and violations of the spatial decoupling, since the subscriber needs to know the identity of the publisher of a received notification in order to perform the signature verification. Such issues are further exacerbated within the context of the IIoT, since we have to consider their resource-constrained nature and the need of minimizing the energy drained from the battery, which imply the need of storing a limited amount of information, executing simple mathematical operations and keeping the amount of additional data to be exchanged (security overhead) as smaller as possible. This strongly limits the applicability of the basic digital signature schemes and calls out for solutions tailored to provide scalable and anonymous signature management with contained resource usage.

D. Our Contribution

The primary objective of this work has been to apply group-based signature [18] to IIoT communications by integrating it within the event-driven publish/subscribe framework presented in [6]. Such a cryptographic technique consists of having the following:

1) only the members of a given group able to sign the messages exchanged within the group;
2) the destinations able to verify if a signature is valid, without disclosing the true identity of the signer;
3) the signature able to be "opened" so as to reveal the identity of the group member that has signed the message.

Specifically, the publish/subscribe service implements a node-clustering scheme based on their specific topic and dynamically elects the cluster head, which is responsible for generating the group key, maintaining the membership information and disseminating the key among the cluster members by using identity-based encryption (IBE) and signature. Despite offering those capabilities, such a solution is known to be inefficient, and some recent works, such as [19] and [20], have been devoted to the scope of resolving such an issue by reducing the signature length and signature creation/verification time in order to contain the latency introduced. We have based our work on these solutions for short group signatures and applied them to the case of event notification within the IIoT. Also, in [21] and [22], group signature is advocated as a promising solution based on qualitative considerations, but not implemented or quantitatively assessed within the context of the IIoT. Concrete usages of group signatures within the IIoT can be found in [23] and [24], and we differ from them since we introduce clustering in order to improve the scalability and efficiency of such a scheme when dealing with a large number of dynamic nodes and an effective setup, thanks to the use of identity-based cryptoprimitives. Therefore, the major contributions of this work are the following ones.

1) We present an analysis of the available platforms for event notification within the IIoT and discuss their solutions to provide the integrity of the exchanged notifications.
2) We have applied the group signature scheme within the context of the infrastructureless publish/subscribe protocol from [6] in order to achieve a scalable and anonymous signature management framework.
3) We used identity-based cryptoprimitives to allow the cluster head to disseminate new group keys and designed a mechanism to revoke group keys when novel members join the clusters and/or some participants leave them.
4) We have performed a simulation-based assessment of the proposed solution in order to present its achievable quality in terms of latency and energy consumption.

E. Roadmap

The rest of this paper is structured as follows. Section II introduces the key aspects of event-based and secure communications within the context of the IIoT by paying attention to the integrity needs of the IIoT. Section III presents the proposed

group key-based solution and its application within an infrastructureless publish/subscribe service. Section IV illustrates the results achieved by running our solution in a simulated scenario. We conclude with Section V, where the lesson learnt and the plan for future work are discussed.

II. Background and Related Work

A. Publish/Subscribe Services Within the IoT

The publish/subscribe interaction pattern [4] perfectly models event-driven communications occurring at the edge of the IIoT among the sensing nodes and the gateways toward the fog computing devices and/or the applications hosted within the cloud. It is an evolution in the distributed producer/consumer system design, characterized by producer applications, called publishers, where events can occur and notifications are generated with a description of those events, and consumer applications, called subscribers, that receive notifications of the events they are interested in. Within this pattern, we have the presence of an abstraction for gluing together publishers and subscribers that deals with the routing of the notifications from the emitting publishers to the receiving subscribers based on their own interests manifested by means of subscriptions, i.e., proper predicates on the context, the type, or the topic of the exchanged notifications. Such an abstraction can be concretely implemented, at the middleware level, by means of brokers, which are applications running on special nodes (which differ from the ones hosting publishers and subscribers since being characterized by a higher amount of computing and storage capabilities and/or available energy), or in a promiscuous manner by having the publishing and subscribing applications (and hence nodes) to deal with the routing of notifications by autonomously establishing an overlay communication infrastructure among themselves. Despite, in the typical solutions for publish/subscribe services, the use of brokers is appreciated for scalability, maintainability, usability, and availability needs, in the specific IIoT scenario, brokers' deployment becomes a serious concern due to the higher economic costs of resulting infrastructure and to the necessity of a pre-optimized planning of the location of the sensors with respect to the available brokers. On the contrary, having a brokerless solution is strongly preferable since it implies reduced costs and deployment efforts as well as it is more adaptive to mobility patterns and scalable with the number of IoT devices. However, the downside is represented by the complexity of managing the overlay among the nodes.

For these reasons, the currently available solutions for event notifications within the IoT, which are facing a large application also within the context of the IIoT, rely on standards where architectures based on the brokers are preferred [12], [25]. First, the OMG issued the Data Distribution Service (DDS) [26] specification for a brokerless event notification, and its adaptation to the peculiarities of the IoT has been proposed in [27] and [28]. Second, the IETF has issued a set of specifications named as Extensible Messaging and Presence Protocol (XMPP) [29], where the XEP-0174 specification [30] has been thought specifically for the IoT, since no intermediaries are needed. Third, there is an ISO standard (ISO/IEC PRF 20922) named Message Queuing Telemetry Transport (MQTT) [31] with an extension known as MQTT for Sensor Networks [32] that is a lightweight broker-based protocol for resource-constrained devices, such as the ones used in the IoT. Finally, the IETF has standardized the Constrained Application Protocol (CoAP) [33] for the web transfers based on the Representational State Transfer (REST) on top of HTTP functionalities, with the possibility of using an optional extension [34] for group communications with IP multicast or multiple unicast sessions. Recently, such an RFC evolved in [35], which defines a broker-based architecture for the CoAP implemented in [36]. Within the academic literature, we can find some proposals for a promiscuous publish/subscribe service, such as the aforementioned protocol presented in [6], providing automatic discovery of newly activated devices and establishment of overlay links among nodes without any broker. This approach will be used as the basis for our proposal.

B. Integrity in the Event Notification Within the IoT

Within the context of publish/subscribe-based event notification, integrity refers to the protection from any possible malicious manipulation of the notification content. Such manipulations may take place on forwarders along the path from the publisher to an interested subscriber, maliciously changing the data contained in the notifications, or on compromised nodes replaying forged notifications by masquerading themselves as legitimate publishers. Digital signature and hashing schemes represent the widely accepted solution for providing such a fundamental security feature.

The existing standards for publish/subscribe services within the IoT provide proper solutions to support integrity demands. First, the OMG has fully standardized the security features for DDS, where the cryptographic service plugin supports all cryptographic operations including digital signatures inserted within the RTPS header. Despite describing why and how using digital signatures, the standard does not indicate which specific technique has to be used. The main products available implement such a standard by adopting state-of-the-art solutions; for a concrete example, Connext DDS Secure from RTI [37] uses the X.509 [38] certificates with a preconfigured shared Certificate Authority, while the signatures are computed with a digital signature algorithm (DSA) [39]. Differently, the OpenSplice framework uses R. Rivest, A. Shamir and L. Adleman (RSA) [39] signatures. Second, with respect to the XMPP, there is a specific extension for signatures called Encapsulated Digital Signatures in XMPP (XEP-0290) [40], which describes a signature approach based upon XML Signatures (XMLDSIG) [41]. Third, in the MQTT context, notifications can contain a digital signature of the contents implemented by using X509 client certificates. The specific technique to be used for computing the signature is not fully standardized, and in [42], the authors propose the use of RSA and a solution based on Elliptic Curves (ECCSA), which represents a valuable signature scheme compared to traditional schemes (RSA and DSA), since it exhibits an equivalent security degree with smaller key sizes, lower complexity, and, hence, faster computation [43]. Finally, there is an on-going work on the security for COAP [44] with a focus on integrity protection

based on JSON Web Signature [45], while Nguyen and Iacono [46] propose a RESTful CoAP message signature generation and verification scheme. These experiences show how the signatures can be integrated within the overall CoAP architecture and in the structure of the exchanged messages, without indicating a given signature approach. Some research efforts aim at fulfilling such a lack, such as [47], where ECCSA is applied, or [48], where EdDSA [49], a variant of Schnorr signature based on Twisted Edwards curves, is recommended.

Despite the various solutions proposed in available products and standards, the literature regarding secure publish/subscribe services lacks of focus on the specific peculiarities of such a kind of approach, and the typical strategy for introducing security services is to adopt schemes taken from secure unicast communications by adapting them to group communication scenarios. This causes three main problems: issues in managing certificates, identity exposure, and scalability limitations. First of all, signatures are encrypted by using a proper encryption key and are verifiable only by using a related decryption key. Typically, a key is a random string, unrelated to the signer identity; therefore, a certification authority is needed in order to bound the adopted cryptographic keys to the user identity. A destination needs to achieve the signer certificate, to check its validity, and to get the signer public key for verifying the signature of the received notification. This causes overheads and inefficiencies, which can be overwhelming within the case of the IoT due to the large number of nodes (whose certificates are needed) and limited storage capacity and availability of battery power, which can be easily drained by continuously acquiring certificates. The problem of managing certificates can be resolved by using identity-based cryptosystems [50], where the public key of a user is easily computable from a string corresponding to the user identity by means of bilinear pairings [51], and without requiring a certification authority. Since the seminal work in [52] that introduced certificateless signatures, a series of papers, such as [53] and [54], have been proposed in order to further improve such a scheme and to make it more secure, by removing the key escrow problem, or more efficient by removing bilinear pairing (whose computations are heavier than the ones in traditional schemes) and basing the signature on the most efficient RSA. However, the use of identity-based cryptosystems is not advantageous in publish/subscribe services. In fact, signature schemes with or without certificates are characterized by the problem of the publisher's identity being exposed during signature verification by a subscriber demanding the public keys of all the interacting publishers. This violates the spatial decoupling property of the publish/subscribe services, since the identity of the publisher needs to be explicit and the event dissemination is no longer anonymous. Moreover, the need for subscribers to know the public keys of the signers still reduces the scalability of the signature scheme.

III. Group Signatures for the Event Notification Within the IIoT

Fig. 3 shows our envisioned approach for the signature of exchanged notifications without violating the anonymity of the

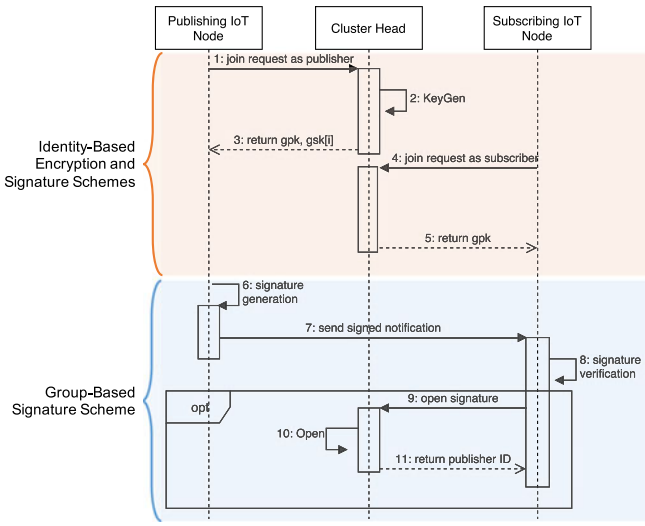


Fig. 3. Sequence diagram of the proposed signing approach.

publish/subscribe service and by guaranteeing the scalability of the communication. The core functionality consists in a new way for authenticating notifications without leaking signer identity, and a suitable approach is a Group Signature scheme [18], which exhibits the following three properties.

1) Only members of a given group, in our case the ones advertised on the same topic, can sign the exchanged notifications.
2) Subscribers can verify if the signature is valid, without disclosing the true identity of the publishing signer.
3) The signature can optionally be "opened" so as to reveal the identity of the group member that has signed the message.

A first practical solution to realize group signatures has been presented in [55] by using dynamic accumulators. However, the inefficiency of the available schemes has limited the widespread adoption of these group signature schemes. This issue has been considered in [19] and [20] in order to reduce the signature length, revocation capability, and signature creation/verification time. A simplified group signature scheme has been proposed in [56] named *Ring Signature* scheme, where the signature creation and verification process is not assigned to a manager, but directly performed by the interested applications. The absence of managers allows the achievement of greater efficiency. In our approach, we have considered the short group signature [19] solution for its simplicity and efficiency, where the generation and management of group signatures is delegated to the cluster head dynamically elected by the publish/subscribe service. However, at the beginning of the approach, the cluster head has to distribute the needed information so that publishers can generate the signatures and the subscribers verify signatures extracted from the received notifications. Such a protection can only be obtained by encrypting and authenticating the messages exchanged by the cluster head with the other members. Also, in this case, it is necessary to adopt a PKI for the management and verification of certificates used for message authentication, causing overheads and inefficiencies. A suitable solution for

simplifying key management and managing certificates is the adoption of identity-based cryptosystems [50], where the public key of a user is easily computable from a string corresponding to the user's identity by means of bilinear pairings [51], and without requiring a certification authority. In the rest of this section, these three aspects of our solution will be described in detail.

A. Group Signature Scheme

The scheme from [19] is made of four distinct algorithms.

1) *KeyGen* deals with generating the key that publishers must use in order to sign their outgoing notifications. It takes as input a parameter n, the number of members authorized to sign, and proceeds as follows. First, it builds two random generators $g_1 \in \mathbb{G}_1$ and $g_2 \in \mathbb{G}_2$, and generates a random number $h \in \mathbb{G}_1 \setminus \{1_{\mathbb{G}_1}\}$, and two other random numbers $\xi_1, \xi_2 \in Z_q^*$. Based on such numbers, it determines $u, v \in \mathbb{G}_1$ as follows: $u = \xi_1^{-1} h$ and $v = \xi_2^{-1}$ so that $u^{\xi_1} = v^{\xi_2} = h$. Then, a random number $\gamma \in Z_q^*$ is generated and $w = g_2^{\gamma}$ is determined. For each ith member of the group that intends to publish notifications and asks the needed information for the consequent signature generation, this algorithm computes a couple (A_i, x_i), where $x_i \in Z_q^*$ is a random number, while $A_i = g_i^{\frac{1}{(\gamma + x_i)}}$. Such a couple corresponds to the private key, namely $gsk[i]$, to be sent to the ith member, and it is stored together with the member's identity by the cluster head so as to be able to open a signature and, consequently, reveal the publisher identity to the subscriber that has requested it. The public key of the group is $gpk = (g_1, g_2, h, u, v, w)$, while the private key of the cluster head is $gmsk = (\xi_1, \xi_2)$.
2) *Sign* describes how to generate a signature for a given notification. Given a public group key gpk, a private publisher key $gsk[i]$, and a notification $M \in \{0,1\}^*$, the notification signature is obtained as follows. A series of random number is generated: $\alpha, \beta, r_\alpha, r_\beta, r_x, r_{\delta_1}, r_{\delta_2} \in Z_q^*$. A series of values are determined: $T_1 = u^\alpha, T_2 = v^\beta$, $T_3 = Ah^{\alpha+\beta}, R_1 = u^{r_\alpha}, R_2 = v^{r_\beta}, R_3 = e(T3, g_2)^{r_x} \cdot e(h,w)^{-r_\alpha - r_\beta} \cdot e(h, g_2)^{-r_{\delta_1} - r_{\delta_2}}, R_4 = T_1^{r_x} \cdot u^{-r_{\delta_1}}$, and $R_5 = T_2^{r_x} \cdot v^{-r_{\delta_2}}$. It evaluates a challenge $C = H(M, T_1, T_2, T_3, R_1, R_2, R_3, R_4, R_5) \in Z_q^*$, where $H(\cdot)$ is a one-way hash function: $\{0,1\}^* \rightarrow Z_q^*$. Then, the following values are estimated: $s_\alpha = r_\alpha + c\alpha, s_\beta = r_\beta + c\beta, s_x = r_x + cx, s_{\delta_1} = r_{\delta_1} + c\delta_1, s_{\delta_2} = r_{\delta_2} + c\delta_2$, and the signature is the following one: $\sigma = (T_1, T_2, T_3, c, s_\alpha, s_\beta, s_x, s_{\delta_1}, s_{\delta_2})$.
3) *Verify* indicates how the signature of a received notification is tested to check its validity. Given the public group key gpk, a notification M, and the relative signature σ, it computes the following values: $R_1^\sim = u^{s_\alpha} \cdot T_1^{-c}, R_2^\sim = T_2^{-c}, R_3^\sim = e(T_3, g_2)^{s_x} \cdot e(h,w)^{-s_\alpha - s_\beta} \cdot e(h, g_2)^{-s_{\delta_1} - s_{\delta_2}} \cdot (\frac{e(T_3,w)}{e(g_1,g_2)})^c, R_4^\sim = T_1^{s_x} \cdot u^{-s_{\delta_1}}, R_5^\sim = T_2^{s_x} \cdot v^{-s_{\delta_2}}$. Then, the result of $H(M, T_1, T_2, T_3, R_1^\sim, R_2^\sim, R_3^\sim, R_4^\sim, R_5^\sim)$ is compared with the challenge c contained in the signature, only if they are equal then the signature is valid.
4) *Open* is responsible for returning the publisher's identifier, given a signature σ and its relative notification, the public group key, and the private cluster head key. First of all, it performs the Verify procedure to check the validity of the signature for the given notification. Considering the first three elements (T_1, T_2, T_3) as a linear encryption, it recovers the value A, i.e., the first element of the publisher private key. Based on the computed A, the cluster head queries its list of publishers and returns the identifier of the corresponding entry.

B. Identity-Based Cryptoprimitives

When the cluster head has to return blocks of data to the requesting member, as indicated in the first two interactions within Fig. 3, the content of those messages must be protected against a malicious adversary that is interested in leaking it. To this aim, we exploit IBE scheme from [57], composed of four procedures.

1) *Setup*: Given a security parameter $k \in Z^+$, it generates a prime number q and three groups $\mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T$ of order q, and a bilinear mapping $e : \mathbb{G}_1 \times \mathbb{G}_2 \rightarrow \mathbb{G}_T$, and selects a random generator $P \in \mathbb{G}_1$ and a random number $s \in Z_q^*$. It computes $P_{pub} = sP$ and chooses two cryptographic hash functions $H_1 : \{0,1\}^* \rightarrow \mathbb{G}_2^*$ and $H_2 : \mathbb{G}_T \rightarrow \{0,1\}^n$. The public parameters for the scheme are $< q, \mathbb{G}_1, \mathbb{G}_2, \mathbb{G}_T, e, P, P_{\text{pub}}, H_1, H_2 >$, while s is the master key to be kept secret.
2) *Extract*: Given the string identifying a node $ID \in \{0,1\}^*$, $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$, and $d_{ID} = sQ_{ID}$, where the last value corresponds to the private key, while ID is the public key.
3) *Encrypt*: Given the message M and the identifier of its destination ID, after determining a random number $\sigma \in \{0,1\}^n$, $Q_{ID} = H_1(ID) \in \mathbb{G}_2^*$ and $g_{ID} = e(P_{\text{pub}}, Q_{ID}) \in \mathbb{G}_T$, the following couple of values is computed: $U = rP$ and $V = M \oplus H_2(g_{ID}^r)$, which is the return of the procedure.
4) *Decrypt*: Given a cyphered message $C =< U, V, M >$, $M = V \oplus H_2(e(U, d_{ID}))$.

It is worth noticing that instead of the symmetric pairing of the original scheme, asymmetric ones have been inserted, and such a choice is motivated by the fact that asymmetric pairing is more secure and efficient than the symmetric one, as proved in [58]. In order to authenticate the messages exchanged by the cluster head without having to manage certificates and incurring in the drawbacks of the PKI, an identity-based signature (IBS) scheme is used, according to [59].

1) *Setup* and *Extract* are executed as in the IBE scheme.
2) *Sign*: Given the private key generated in the previous procedure, it is used to encrypt the message, which is assumed as the signature.
3) *Verify*: Given the public key of the message sender, the signature is decrypted and the result is compared with the

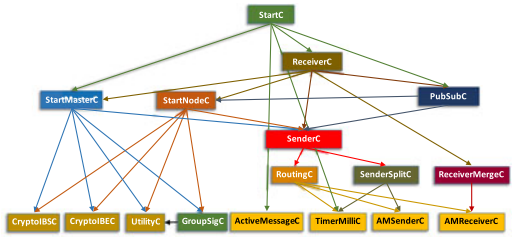


Fig. 4. Components of the proposed prototype.

content of the message. If they match, then the signature is verified.

Only the setup parameters of the group signature schemes are exchanged by the cluster head with the other members, while the same needed data for the IBE and IBS schemes can be computed once and preloaded within each IoT nodes before deploying them.

IV. Empirical Assessment

The security of the short group signature scheme adopted in this paper has been proved in [19] in the random oracle model, and the use of asymmetric pairing and preloaded setup data at each sensor node in the identity-based cryptoprimitives used to exchange the signature and verification parameter from the cluster head to the publishers and subscribers guarantees a high degree of security, as demonstrated in the literature, such as in [58]. Since the approach is theoretically secure, we need to show its quality in terms of exhibited performance and energy consumption.

To this aim, the proposed approach has been implemented for sensors based on the TinyOS operative system [60], which allows one to implement applications by using a component-based event-driven programming language called nesC and based on the widely known C language. We have tested our application by using the TOSSIM simulator [61], while for the cryptographic operations, we have used the Relic library [62], which focuses on the efficiency and flexibility of the pairing operations, making them suitable for the resource-constrained nodes composing the IoT. Fig. 4 schematically illustrates the implementation of our prototype, which is organized in layers, with components sending commands and requests to the ones below them, which in turn notify events to the requesting components at the higher layer. StartC is the root component of the application, which instantiates and starts the other ones. All the nodes have the same internal architecture, but based on the assumed role (publisher, subscriber, or cluster head), it triggers the opportune functionalities provided by the components. Specifically, StartMasterC is the component encapsulating the application logic of the cluster head by managing the cluster members and the public parameters for the group signature, while StartNodeC is the component containing the cluster member logic to obtain the publish parameters from the cluster head and make signature generation and/or verification. SenderC and ReceiverC are the two communication endpoints for exchanging notifications and can be supported by the SenderSplitC and ReceiverMergeC when a notification

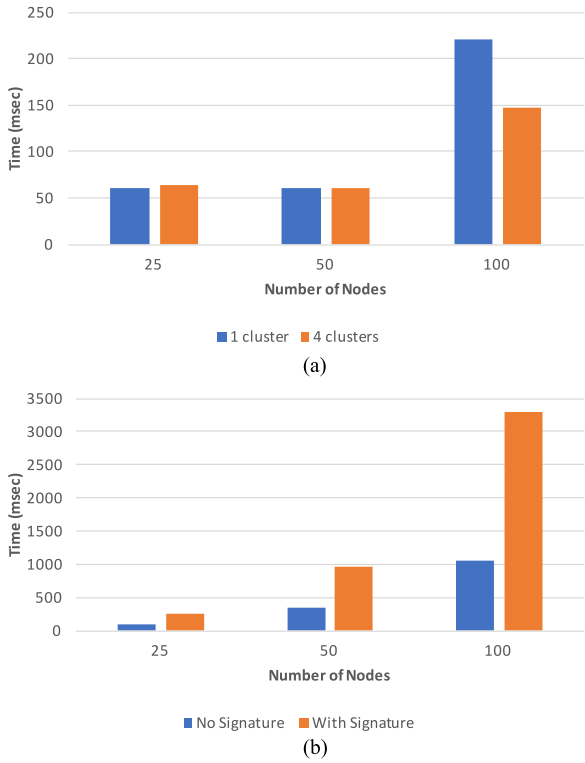


Fig. 5. (a) Time to join a cluster. (b) Time to publish a notification.

exceeds the maximum size of a packet and must be fragmented. PubSubC realize the operations to publish and subscribe to a certain kind of events. At the lowest level of our prototype, we have a set of components that implements the needed operations of group signature, IBS, and cryptography described in the previous section, in addition to the low-level operation of message serialization, wireless connection establishment, and recovery or timing.

A series of experiments with a varying number of nodes (respectively, 25, 50, and 100 nodes) has been performed by running our prototype in TOSSIM, by repeating each test case five times and reporting the mean over the obtained measures of merit. The assumed hardware for the nodes is Micaz, which is equipped with an 8-bit Atmel AVR microcontroller, with a 4-kB RAM, a programmable flash memory of 128 kB, a secondary memory of 512 kB, and a radio chip cc2420 with a 250-kb/s data rate. We have tested the following operations: joining a group, generating a group signature, verifying a received group signature, tracing of a signing entity, and publish (and relative consuming) operations. Fig. 5 shows the obtained results in terms of the mean time needed to complete these operations, while Fig. 6 represents the average energy consumption of the most important one, i.e., the publication and relative reception of a notification.

Fig. 5(a) illustrates the time needed to join a cluster, which consists of a request to the cluster head, and a response with the needed data to perform the signature by the publisher. Such an operation is done right after the cluster head has been elected. In the figure, it is possible to notice two different configurations of the publish/subscribe service: one with no clustering, i.e., all the

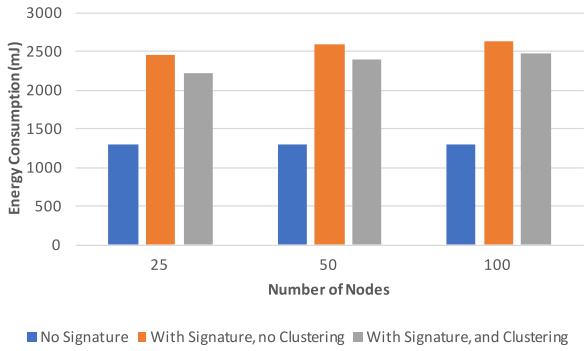


Fig. 6. Energy consumption of a forwarding operation.

nodes are implicitly within a single cluster managed by a given cluster head, and the other with the overall nodes clustered in four groups. In these two configurations, the join time is almost identical until 50 nodes, while in the last case, we have a slight divergence among the two configurations. This means that until a given number of cluster participants, the cluster head is able to cope with the incoming requests, while when the number of node is excessive, the cluster head starts to be overwhelmed by the requests and starts to slow down. Clustering helps to alleviate such a performance issue, as evident in the figure. The generation of the signature done by each publisher before passing a notification to the communication protocol takes on average 13 ms, while the dual operation of signature verification is faster and lasts for about 7 ms. In the case a subscriber wants to trace the identity of the publisher that has generated a received notification by contacting the cluster head, 2 ms are needed so as to let the cluster head identify the right publisher identity based on the received signature. The overall completion time for this operation must consider the delay of sending the request and obtaining a reply, which depends on the network behavior and the path length to the cluster head, in addition to the time for the identity inference. Fig. 5(b) indicates the overall time elapsed from the publish operation to the consumption of the notification by the subscribing application. In the figure, we have compared this operation without and with our signing approach. As expected, the use of our approach implies a performance worsening, which increases when the number of nodes grows (the prototype has been configured with four clusters so as to limit the signing costs). The first reason is that the insertion of a signature implies an increase in the notification size from 5 to 296 byte, which is difficult to limit, since the robustness of our approach to possible attacks depends on the length of the signature. Moreover, every time a node receives a notification, even if not being interested and acting as a forwarder, it must verify the attached signature, and this has a performance cost. When the network grows, the number of hops to reach a destination probably augments, causing the trend illustrated in the figures. Moreover, the increasing size of a notification causes the need of fragmenting it in multiple packets and managing the needed reassembling of the overall fragments and their retransmission in case of losses, implying the consequent increase of the delivery time. A last consideration is related to the energy consumption, illustrated in Fig. 6, where a signing scheme augments it, but clustering is able to slightly reduce such a cost, since it depends not only on the exchanged messages, but also on the mathematical computations performed by the nodes when generating or verifying signatures.

V. Conclusion and Final Remarks

In this paper, we have presented the known issues associated with ensuring message integrity and authentication by means of digital signatures within the context of publish/subscribe services. The currently available solutions lack energy efficiency and scalability, which are fundamental requirements within the context of the IIoT; moreover, they violate the anonymity and decoupling properties for the event notification in publish–subscribe schemes. To cope with these problems, we have proposed a group-signature-based scheme and applied it to a prototype of infrastructureless topic-based publish/subscribe service for sensors. We have empirically assessed it so as to measure the consequent performance worsening and the increase in the battery consumption. An open issue in our approach is the key revocation, mainly related to a publisher leaving the group. In our approach, we have adopted the simple solution from [63], where the signing and verification parameters, respectively, gpk and $gsk[i]$, for the ith publisher and gpk for the subscribers are changed and retransmitted when a node leaves. Despite having a simple implementation, such a solution is not optimal since the associated cost (in terms of revocation time and energy consumption) is considerable. As a future work, we will investigate more suitable revocation schemes among the ones in the current literature and adapt it in our approach; in addition, other signature schemes suitable for our aims, such as batch signatures [64] or ring-based ones [65], will be studied.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Comput. Netw.*, vol. 54, no. 15, pp. 2787–2805, 2010.

[2] A. V. Dastjerdi and R. Buyya, "Fog computing: Helping the Internet of Things realize its potential," *Computer*, vol. 49, no. 8, pp. 112–116, Aug. 2016.

[3] W. Feng, Z. Yan, H. Zhang, K. Zeng, Y. Xiao, and T. Hou, "A survey on security, privacy and trust in mobile crowdsourcing," *IEEE Internet Things J.*, 2017, doi: 10.1109/JIOT.2017.2765699 .

[4] P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surveys*, vol. 35, no. 2, pp. 114–131, Jun. 2003.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Fourth Quarter 2015.

[6] C. Esposito, A. Castiglione, F. Palmieri, M. Ficco, and K. K. R. Choo, "A publish/subscribe protocol for event-driven communications in the internet of things," in *Proc. IEEE 14th Int. Conf. Dependable, Auton. Secure Comput.*, Aug. 2016, pp. 376–383.

[7] Y. Koren, *The Global Manufacturing Revolution: Product-Process-Business Integration and Reconfigurable Systems*. New York, NY, USA: Wiley, Jun. 2010.

[8] M. Wollschlaeger, T. Sauter, and J. Jasperneite, "The future of industrial communication: Automation networks in the era of the Internet of Things and Industry 4.0," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 17–27, Mar. 2017.

[9] EFFRA, "Factories of the Future—Multi-annual roadmap for the contractual PPP under Horizon 2020," Oct. 24, 2016. [Online]. Available: https://ec.europa.eu/research/industrial_technologies/factories-of-the-future_en.html

[10] Executive Office of the President President's Council of Advisors on Science and Technology, "Report to the President on Capturing Domestic Competitive Advantage in Advanced Manufacturing," Oct. 24, 2016. [Online]. Available: http://energy.gov/eere/downloads/report-president-capturing-domestic-competitive-advantage-advanced-manufacturing

[11] F. Tao, Y. Zuo, L. D. Xu, and L. Zhang, "IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing," *IEEE Trans. Ind. Informat.*, vol. 10, no. 2, pp. 1547–1557, May 2014.

[12] L. D. Xu, W. He, and S. Li, "Internet of Things in industries: A survey," *IEEE Trans. Ind. Informat.*, vol. 10, no. 4, pp. 2233–2243, Nov. 2014.

[13] S. Jeschke, C. Brecher, T. Meisen, D. Özdemir, and T. Eschert, "Industrial Internet of Things and cyber manufacturing systems," *Industrial Internet of Things*. Cham, Switzerland: Springer, 2017, pp. 3–19.

[14] S. Mumtaz, A. Alsohaily, Z. Pang, A. Rayes, K. F. Tsang, and J. Rodriguez, "Massive Internet of Things for industrial applications: Addressing wireless IIOT connectivity challenges and ecosystem fragmentation," *IEEE Ind. Electron. Mag.*, vol. 11, no. 1, pp. 28–33, Mar. 2017.

[15] M. Nawir, A. Amir, N. Yaakob, and O. B. Lynn, "Internet of Things (IoT): Taxonomy of security attacks," in *Proc. 3rd Int. Conf. Electron. Des.*, Aug. 2016, pp. 321–326.

[16] M. O'Brien and G. R. S. Weir, "Understanding digital certificates," in *Proc. 2nd Int. Conf. Cybercrime Forensics Edu. Training*, Sep. 2008.

[17] C. Esposito and M. Ciampi, "On security in publish/subscribe services: A survey," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 2, pp. 966–997, Second Quarter 2015.

[18] D. Chaum and E. Heyst, "Group signatures," in *Proc. 10th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1991, vol. 547, pp. 257–265.

[19] D. Boneh, X. Boyen, and H. Shacham, "Short group signatures," in *Annu. Int. Cryptol. Conf.*, 2004, vol. 3152, pp. 41–55.

[20] S. Zhou and D. Lin, "Group signatures with reduced bandwidth," *IEE Proc.—Inf. Security*, vol. 153, no. 4, pp. 146–152, Dec. 2006.

[21] H. Yue, L. Guo, R. Li, H. Asaeda, and Y. Fang, "DataClouds: Enabling community-based data-centric services over the Internet of Things," *IEEE Internet Things J.*, vol. 1, no. 5, pp. 472–482, Oct. 2014.

[22] L. Malina, J. Hajny, R. Fujdiak, and J. Hosek, "On perspective of security and privacy-preserving solutions in the internet of things," *Comput. Netw.*, vol. 102, no. Suppl. C, pp. 83–95, 2016.

[23] J. Su, D. Cao, B. Zhao, X. Wang, and I. You, "ePASS: An expressive attribute-based signature scheme with privacy and an unforgeability guarantee for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 33, no. Suppl. C, pp. 11–18, 2014.

[24] C. Lai, H. Li, X. Liang, R. Lu, K. Zhang, and X. Shen, "CPAL: A conditional privacy-preserving authentication with access linkability for roaming service," *IEEE Internet Things J.*, vol. 1, no. 1, pp. 46–57, Feb. 2014.

[25] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of Things: A survey on enabling technologies, protocols, and applications," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 4, pp. 2347–2376, Fourth Quarter 2015.

[26] OMG, "Data Distribution Service (DDS) for Real-Time Systems, v1.2," Sep. 2012. [Online]. Available: www.omg.org

[27] A. Hakiri, P. Berthou, A. Gokhale, and S. Abdellatif, "Publish/subscribe-enabled software defined networking for efficient and scalable IoT communications," *IEEE Commun. Mag.*, vol. 53, no. 9, pp. 48–54, Sep. 2015.

[28] A. Hakiri, A. Gokhale, P. Berthou, D. Schmidt, and T. Gayraud, "Software-defined networking: Challenges and research opportunities for future internet," *Comput. Netw.*, vol. 75, pp. 453–471, Dec. 2014.

[29] IETF, "RFC 6120: Extensible Messaging and Presence Protocol (XMPP)," Mar. 2016. [Online]. Available: http://tools.ietf.org/html/rfc6120

[30] P. Saint-Andre, "XEP-0174: Serverless Messaging," Mar. 2016. [Online]. Available: http://www.xmpp.org/extensions/xep-0174.html

[31] D. Locke, "MQ Telemetry Transport (MQTT) V3.1 Protocol Specification," Mar. 2016. [Online]. Available: http://www.ibm.com/developerworks/webservices/library/ws-mqtt/

[32] A. Stanford-Clark and H. L. Truong, "MQTT for sensor networks (MQTT-S)," Mar. 2016. [Online]. Available: http://www.mqtt.org/MQTTs_Specification_V1.0.pdf

[33] C. Bormann, A. P. Castellani, and Z. Shelby, "CoAP: An application protocol for billions of tiny internet nodes," *IEEE Internet Comput.*, vol. 16, no. 2, pp. 62–67, Mar. 2012.

[34] A. Rahman and E. Dijk, "Group communication for the constrained application protocol (CoAP)," Internet Eng. Task Force (IETF), Request Comments: 7390, Oct. 2014. [Online]. Available: https://tools.ietf.org/html/rfc7390

[35] M. Koster, A. Keranen, and J. Jimenez, "Publish-subscribe broker for the constrained application protocol (CoAP)," Netw. Working Group, Internet Eng. Task Force (IETF), Internet-Draft, Mar. 2017. [Online]. Available: https://tools.ietf.org/html/draft-ietf-core-coap-pubsub-01

[36] M. Kovatsch, S. Duquennoy, and A. Dunkels, "A low-power CoAP for contiki," in *Proc. IEEE 8th Int. Conf. Mobile Ad-Hoc Sens. Syst.*, Oct. 2011, pp. 855–860.

[37] RTI. Connext DDS Secure, Jul. 2017. [Online]. Available: https://www.rti.com/products/secure

[38] R. Housley, W. Polk, W. Ford, and D. Solo, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Jul. 2013. [Online]. Available: http://www.ietf.org/rfc/rfc3280.txt

[39] W. Stallings, *Network Security Essentials—Applications and Standards*, 4th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.

[40] K. Zeilenga, XEP-0290: Encapsulated Digital Signatures in XMPP, 2011. [Online]. Available: https://xmpp.org/extensions/xep-0290.html

[41] D. Eastlake, J. Reagle, D. Solo, F. Hirsch, and T. Roessler, *XML Signature Syntax and Processing: W3C Recommendation*, 2nd ed., Jul. 2013. [Online]. Available: http://www.w3.org/TR/xmldsig-core/

[42] A. Mektoubi, H. L. Hassani, H. Belhadaoui, M. Rifi, and A. Zakari, "New approach for securing communication over MQTT protocol: A comparison between RSA and elliptic curve," in *Proc. 3rd Int. Conf. Syst. Collaboration*, Nov. 2016, pp. 1–6.

[43] N. Koblitz, A. Menezes, and S. Vanstone, "The state of elliptic curve cryptography," *Des., Codes Cryptography*, vol. 19, pp. 103–123, 2000.

[44] J. Mattsson, G. Selander, and L. Seitz, Object security for COAP, 2014. [Online]. Available: https://www.ietf.org/proceedings/91/slides/slides-91-ace-2.pdf

[45] M. Jones, J. Bradley, and N. Sakimura, JSON Web Signature (JWS), 2015. [Online]. Available: https://tools.ietf.org/html/rfc7515

[46] H. V. Nguyen and L. L. Iacono, "REST-ful CoAP message authentication," in *Proc. Int. Workshop Secure Internet Things*, Sep. 2015, pp. 35–43.

[47] Lavanya and Natarajan, "Lightweight authentication for COAP based IOT," in *Proc. 6th Int. Conf. Internet Things*, 2016, pp. 167–168.

[48] M. Tiloca, G. Selander, and F. Palombini, Secure group communication for CoAP—draft-tiloca-core-multicast-oscoap-03, Jul. 2017. [Online]. Available: https://ericssonresearch.github.io/Multicast-OSCOAP/draft-tiloca-core-multicast-oscoap.html

[49] S. Josefsson and I. Liusvaara, Edwards-Curve Digital Signature Algorithm (EdDSA), Jul. 2017. [Online]. Available: https://tools.ietf.org/html/rfc8032

[50] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1985, vol. 196, pp. 47–53.

[51] F. Zhang, R. Safavi-Naini, and W. Susilo, "An efficient signature scheme from bilinear pairings and its applications," in *Proc. Int. Workshop Public Key Cryptography*, 2004, vol. 2947, pp. 277–290.

[52] S. S. Al-Riyami and K. G. Paterson, "Certificateless public key cryptography," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2003, vol. 2894, pp. 452–473.

[53] R. Tso, X. Huang, and W. Susilo, "Strongly secure certificateless short signatures," *J. Syst. Softw.*, vol. 85, no. 6, pp. 1409–1417, Jun. 2012.

[54] J. Zhang and J. Mao, "An efficient RSA-based certificateless signature scheme," *J. Syst. Softw.*, vol. 85, no. 3, pp. 638–642, Mar. 2012.

[55] J. Camenisch and A. Lysyanskaya, "Dynamic accumulators and application to efficient revocation of anonymous credentials," in *Proc. Annu. Int. Cryptol. Conf.*, 2002, vol. 2442, pp. 61–76.

[56] R. L. Rivest, A. Shamir, and Y. Tauman, "How to leak a secret," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Security*, 2001, vol. 2248, pp. 552–565.

[57] D. Boneh and M. Franklin, "Identity-based encryption from the weil pairing," in *Proc. 21st Annu. Int. Cryptol. Conf.*, Aug. 2001, pp. 213–229.

[58] M. S. Kiraz and O. Uzunkol, "Still wrong use of pairings in cryptography," *arXiv preprint arXiv:1603.02826*, 2016.

[59] K. G. Paterson and J. C. N. Schuldt, "Efficient identity-based signatures secure in the standard model," in *Proc. 11th Australas. Conf. Inf. Security Privacy*, Melbourne, VIC, Australia, Jul. 2006, pp. 207–222.

[60] P. Levis *et al.*, "TinyOS: An operating system for sensor networks," *Ambient Intell.*, vol. 35, pp. 115–148, 2005.

[61] P. Levis, N. Lee, M. Welsh, and D. Culler, "TOSSIM: Accurate and scalable simulation of entire TinyOS applications," in *Proc. 1st Int. Conf. Embedded Netw. Sens. Syst.*, 2003, pp. 126–137.

[62] D. F. Aranha and C. P. L. Gouvêa, "RELIC is an efficient library for cryptography." [Online]. Available: https://github.com/relic-toolkit/relic

[63] G. Ateniese, D. Song, and G. Tsudik, "Quasi-efficient revocation of group signatures," in *Proc. 6th Financial Cryptography Conf.*, 2016, vol. 2357, pp. 88–98.
[64] Z. Yan, W. Feng, and P. Wang, "Anonymous authentication for trustworthy pervasive social networking," *IEEE Trans. Comput. Social Syst.*, vol. 2, no. 3, pp. 11–18, Feb. 2016.
[65] J. K. Liu, M. H. Au, W. Susilo, and J. Zhou, "Linkable ring signature with unconditional anonymity," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 1, pp. 157–165, Jan. 2014.