Key Distribution Protocol for Industrial Internet of Things Without Implicit Certificates

Mohamed H. Eldefrawy, Nuno Pereira, and Mikael Gidlund, *Senior Member, IEEE*

***Abstract*—The deployment of the Internet of Things (IoT) in industry, called the Industrial IoT (IIoT), is supporting the introduction of very desirable improvements, such as increasing production flexibility, self-organization, and real-time and quick response to events. However, security and privacy challenges are still to be well addressed. The IIoT requires different properties to achieve secure and reliable systems and these requirements create extra challenges considering the limited processing and communication power available to IIoT field devices. In this research article, we present a key distribution protocol for IIoT that is computationally and communicationally lightweight (requires a single message exchange) and handles node addition and revocation, as well as fast rekeying. The scheme can also resist the consequences of node capture attacks (we assume that captured nodes can be detected by the gateway and previous works have shown this assumption to be acceptable in practice), server impersonation attacks and provides forward/backward secrecy. We show formally the correctness of our protocol and evaluate its energy consumption under realistic scenarios using a real embedded platform compared to previous state-of-the-art key-exchange protocols, to show our protocol reliability for IIoT.**

***Index Terms*—Cyber assurance, Industrial Internet of Things (IIoT), key distribution protocol.**

I. Introduction

INTERNET-CONNECTED devices are already widely used in many industrial settings and, until 2025, it is projected that the largest growth of the Internet of Things (IoT) will be in industrial applications [1]–[3] —the so-called Industrial IoT (IIoT) [4], [5]. Fig. 1 shows the infrastructure of a typical IIoT network, where field devices, often low-power and wirelessly connected, acquire data that is used for actuation and through the plant's information systems. This infrastructure is critical for the continued operation of the plant; its components interact in complex manners and can even have safety-critical properties, as wrongful actuation might endanger lives [5].

Manuscript received May 23, 2018; revised July 20, 2018; accepted August 8, 2018. Date of publication August 14, 2018; date of current version February 25, 2019. This work was supported by grant HYPERLINK "tel:20150367" 20150367 of the Swedish Knowledge Foundation and by grant HYPERLINK "tel:20201010" 20201010 of the European Regional Fund. *(Corresponding author: Mohamed H. Eldefrawy.)*

M. H. Eldefrawy and M. Gidlund are with the Department of Information Systems and Technology, Faculty of Science, Technology, and Media, Mid Sweden University, 851 70 Sundsvall, Sweden (e-mail: mohamed.eldefrawy@miun.se; mikael.gidlund@miun.se).

N. Pereira is with the Computer Engineering Department, School of Engineering (DEI/ISEP), Polytechnic of Porto, 4200-072 Porto, Portugal (e-mail: nap@isep.ipp.pt).

Digital Object Identifier 10.1109/JIOT.2018.2865212

In this paper, we address the problem of establishing secure channels between the several components of this IIoT infrastructure and protect the information flow. We propose a key management scheme that IIoT field devices can use to exchange security credentials used to secure the communication channels. A very important consideration when designing cryptographic schemes for the IIoT is that IIoT field devices have very limited processing, storage, and communication resources. Researchers have recently proposed key management schemes that exploit elliptic curve cryptography (ECC) [6]–[8] as ECC offers more efficient implementations when compared to other public-key algorithms. Nevertheless, ECC primitives have computational and energy costs that are an order of magnitude higher than symmetric key algorithms [9]. Our proposal does not involve public key cryptography, but instead, we only use forward hashing to attain a much more efficient and reliable technique. While our scheme requires repeatedly computing a number of hashes, we show (in Section VI) that the result is still competitive (in terms of resource usage) for resource-constrained IIoT devices when compared to previous works. Similar to previous works (e.g., [6] and [7]), our protocol requires storing a secret on the nodes (in our case, the secret hash seed). Our security depends on the fact that an attacker cannot capture a node and use this secret to derive new keys without being detected. This is a reasonable assumption that has been shown to be practical [10]–[12] (see discussion in Section IV-A). Finally, we note that our protocol makes an extremely efficient use of communication resources, as it requires only one message to be exchanged, and this is a very appealing feature for resource-constrained IIoT networks.

IIoT devices have specific requirements that are not common in the consumer IoT. In the context of this paper on a key-distribution protocol for IIoT, we can identify several relevant distinguishing features of IIoT as follows [14].

1) Bounded and short time response is required in continuous manufacturing processes. These processes are monitored and controlled over comprehensive sensing and actuation systems to be maintained without interruptions in a fast and predictable routine. To satisfy this requirement, we proposed an *instant* key distribution protocol that *requires a single message exchange* to distribute authentic session keys between the network nodes, and therefore, it is predictable and introduces minimal communication overhead.
2) Reliability is a major concern for IIoT. Dissimilar to consumer IoT that works on domestic and residential

Fig. 1. Typical IIoT infrastructure.

areas, IIoT systems are operated in tough and dangerous environments, often with safety implications, that is, a failure of the system may jeopardize lives. To address this aspect, we introduced a simple and repeatable key distribution protocol which, we believe, contributes to the overall reliability of the system, and does not required network sensor nodes to be accessed locally after deployment.

3) Power consumption, IIoT systems need to be run over long periods before considering a battery replacement/charging. This is important to facilitate maintenance and IIoT are more sensitive to such return on investment aspects than consumer IoT. In this paper, we present a lightweight key distribution protocol for the IIoT context with low computational cost operations (only Hashing and Xoring) by avoiding public key cryptography, *in all its forms*, to make it more appropriate and convenient for constrained and limited power devices.
4) Security is a serious problem for all IoT applications; however, industrial uses need more robust and reliable measures due to their safety implications. Our key distribution protocol for IIoT is formally proofed with Scyther security validation tool [13]. Our protocol can compensate the consequences of node capture and resist server impersonation attack. In addition, it provides forward/backward secrecy to assure the concealment of past information and session keys.
5) Scalability, as industrial systems are often composed of thousands (or even more) sensors and actuators, it is necessary to dynamically add and revoke nodes effortlessly to support thousands of new sensors and actuators. In our protocol, we carefully designed a robust method of nodes addition and revocation, as well as the fast rekeying process to be initiated by the gateway (GW) itself, for more information check Section IV-B.

Considering the IIoT specificities, an authenticated key distribution protocol needs to consider the following properties.

1) The CIA triad (i.e., confidentiality, integrity, and availability) is a set of measures undertaken to prevent information from reaching unauthorized users and, conversely, make it available for legitimate parties, and the data exchanged should be intact and unchanged [15].
2) Forward secrecy is a property of secure communication protocols in which long-term keys compromise will not affect past session keys. Forward secrecy protects past sessions against potential attacks of secret keys and/or passwords.
3) Implicit key authentication (of A to B), for an authentication protocol, is to provide implicit key authentication, if B is assured that no other entity aside from A can learn the value of the shared secret key.
4) Key confirmation is the assurance to participants of the key-establishment protocol that the intended recipient of the shared key actually possesses it (i.e., the shared key).
5) Explicit key authentication (EKA), if we have both, the implicit key authentication and the key confirmation, then we can say that our key agreement protocol provides EKA. A key distribution that provides EKA is called authenticated key agreement with key confirmation protocol.
6) Key control, in which none of the principles can make the key to be any predefined value; or else, any party can force the use of old keying factors. We need to assure that each principle does not need to rely on any other party to generate appropriate keys.
7) Key freshness, in which we need to assure that the derived session key is fresh, as against the reusing of old keying material. Our design takes into account these desirable properties. Our scheme is particularly suited for IIoT and takes advantage of the more controlled setup and deployment procedures of industrial networks in a bounded architecture.

Our contributions can be briefed as follows.

1) A proposal of a key distribution protocol for resource-constrained IIoT devices that require a single message exchange.
2) The proposed protocol considers robust nodes addition and revocation, as well as fast rekeying.
3) Our protocol is also robust to node capture and server impersonation attacks.
4) Also, it provides forward/backward secrecy.

Additionally, we formally prove the security of our protocol using the Scyther tool [13], [16], as well as the BAN logic postulate and developed an implementation for real embedded platforms that we use to evaluate its computing and communication costs under realistic assumptions. The evaluation results confirm that our protocol has energy costs that are competitive when compared to previous protocols [6]–[8]. More precisely, our protocol consumes less energy than [8] and is similar to [6] and [7].

The remainder of this paper is organized as follows. Section II shows the background and related work. Section III illustrates our contribution. Section IV, evaluates the security

of our solution. Section V formally proves the anticipated security goal of our protocol. Section VI evaluates the performance of our contribution. Finally, the conclusions will be drawn in Section VII.

II. Background and Related Work

IIoT networks have very restricted requirements in terms of computational power and execution time, unfortunately, regular authenticated group messaging protocols, like [17] consume too many resources for the typical IIoT device. Accordingly, we have to find a way out to achieve the desired security properties with limited resources. This paper relies on two building blocks: 1) hash chains and 2) the Chinese remainder theorem (CRT), which we will detail in the following sections. The one-way hash chain was presented by Lamport [18], also known as S/Key OTP [19]. It is computationally expensive for the end user; hence, it uses backward and finite hash chains. Additionally, Lamport's scheme is vulnerable to some attacks, e.g., small-challenge attack [20], replay attack [21], and man-in-the-middle attack [22]. Elliptic curve Qu–Vanstone (ECQV) implicit certificates [23] are utilized for PKI management protocol for IoT applications as a practical alternative of X.509 certificate [6]–[8], [24], [25] since it is much more efficient and suitable for limited resources networks. Recently, Sciancalepore *et al.* [6] proposed a key management protocol for IIoT based on an ECQV technique [23] for authentication and key negotiation. This is considered an important related work and, in Section II-D, we will discuss [6], including several relevant weaknesses.

A. Lamport's Scheme

Lamport originally presented a technique of hash chains [18]. It applies a one-way hash function $h(\cdot)$ for N times to some initial value named a seed or s to produce a chain of cascaded hashes of length N

$$h^1(s), h^2(s), \ldots, h^{N-1}(s), h^N(s). \tag{1}$$

As a response to a challenge-response authentication for the ith session, user forwards the following answer:

$$k_i(s) = h^{N-i}(s). \tag{2}$$

The authentication server validates the user response using the following equality:

$$h(k_i(s)) \stackrel{?}{=} h^{N-i+1}(s). \tag{3}$$

As this hash content $h^{N-i+1}(s)$ is previously set in the remote servers storage starting with the ith authentication. Because of a valid verification, the remote servers' storage is reloaded with a new hashed image. Lamport has constraints on the verification sessions cycles, after N authentications sessions, it requires restarting the process. Furthermore, it is vulnerable to small challenge attack [20]. In addition, the users are loaded with heavy calculations over the phase of initialization that turns it inappropriate for IIoT networks.

B. Chinese Remainder Theorem

To make it clear to the reader, we will try to investigate the CRT in order to show its simultaneous congruence power. For the integers $m_1, m_2, \ldots, m_l$, if they are pairwise relatively prime, then the following congruent equations:

$$\begin{aligned} x &\equiv s_1 \bmod m_1 \\ x &\equiv s_2 \bmod m_2 \\ &\vdots \\ x &\equiv s_l \bmod m_l \end{aligned}$$

derives a single solution

$$x = \sum_{i=1}^{l} s_i M_i^{-1} M_i \bmod M \tag{4}$$

where

$$M = \prod_{i=1}^{l} m_i \tag{5}$$

$$M_i = \frac{M}{m_i} \tag{6}$$

$$M_i^{-1} M_i \equiv 1 \bmod m_i. \tag{7}$$

We believe that CRT can offer a great inspiration in information hiding and data ciphering [26], [27].

C. CRTBA Broadcast Authentication

As far as we can tell that [28] first introduced the CRT in the broadcast authentication for wireless sensor networks to overcome the limitation of TESLA and mu-TESLA [29], [30]. The protocol is considering three different phases: 1) distribution; 2) message signing; and 3) message authentication phase. All sensors are equipped with a seed k_n, one-way hash $h(\cdot)$, and two dissimilar modules, n_A and n_B, for the CRT, *before deployment*. As soon as the GW broadcasts a new message m to the node for the ith session, it evaluates the message MAC as $M = \mathrm{MAC}_{k_i}(m)$. Afterward, GW encrypts k_i and M using the two private modules n_A and n_B over the CRT to get a simultaneous congruence

$$\begin{aligned} X &\equiv k_i \bmod n_A \\ X &\equiv M \bmod n_B. \end{aligned}$$

Accordingly, X is broadcast. The sensor nodes can recover k_i from X, to apply the one-way hash function $h(\cdot)$ to validate $k_j \stackrel{?}{=} h^{i-j}(k_i)$, where k_j is the former genuine key that sensor nodes have got. Sadly, this scheme has a limitation of using reverse hashing for keys generation.

D. Sciancalepore et al. KMP

Recently an implicitly certificated key agreement protocol [6] has been presented in which to achieve an authenticated key agreement for IIoT networks. However, we have spotted a different weakness of it in terms of security and efficiency. In this section we will try to go through it and show its shortcomings to the reader. This key agreement protocol utilizes a public key encryption in the form of elliptic curve

Diffie–Hellman (ECDH) [31] via the ECQV implicit certificate [23] to achieve a shared session key for any two devices, A and B that need to communicate. Sciancalepore *et al.* [6] assumed that all constrained devices, trust the same certificate authority (CA) and they have their own security credentials in terms of public and private key pairs as well as the CA's public key. In their protocol phase of certificate issuing, they assumed that each communication's peer X (i.e., A and B) produces a positive random integer r_x to calculate the elliptic curve point $R_X = r_x \cdot G$, as G is an order n generator of the elliptic curve group $\mathbb{G}$, to forward it to the CA. Consequently, CA produces a random positive integer k, and returns the implicit certificate[1]

$$C_X = R_X + k \cdot G \tag{8}$$

along with the implicit signature

$$\gamma_x = p_{ca} + k \cdot H(C_X, X) \tag{9}$$

back to X, *as a credentials preloading*. Then X generates its private key

$$p_x = \gamma_x + r_x \cdot H(C_X, X) \tag{10}$$

and its public key

$$P_X = p_x \cdot G. \tag{11}$$

Afterward, in the certificate exchanging phase, again each peer X (i.e., A and B) sends to each other an initial message contains: a nonce ρ_x and an implicit certificate C_X, to allow the other partner to calculate its public key as an elliptic curve point using

$$P_X = P_{CA} + C_X \cdot H(C_X, X) \tag{12}$$

considering that $H(\cdot)$ is a one-way cryptographic hash function. To realize the shared session key K_{AB} between A and B, as A is having C_B, P_B and its private key p_a, it will be $K_{AB} = p_a P_B = p_a p_b G$ as a straightforward ECDH substitution and such is the case for B as well. Then to achieve a message exchange authentication they utilized $\Gamma[x, y]$ which refers to a symmetric authentication algorithm of x and y, *as an HMAC for example*, so according to [6] A and B need to compute their authentication tags

$$\alpha_A = \Gamma[K_{AB}, (C_A, A, C_B, B, \rho_A, \rho_B)] \tag{13}$$

$$\alpha_B = \Gamma[K_{AB}, (C_B, B, C_A, A, \rho_B, \rho_A)]. \tag{14}$$

As soon as the two communication peers authenticate their exchanged messages, they go for the shared key derivation using an ordinary key derivation function χ, as

$$P_k = \chi(K_{AB}, \rho_B, \rho_A). \tag{15}$$

[1]Also known by the ECQV as a public key reconstruction data.

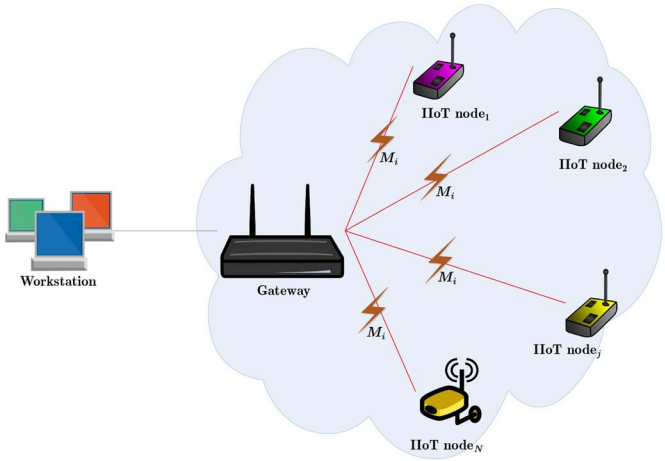

Fig. 2. System model for the proposed key distribution protocol.

1) Drawbacks of Sciancalepore et al.:

a) Node capturing attack: Sciancalepore *et al.* [6], [7] assumed the security credentials (i.e., implicit certificate C_X, implicit signatures γ_x, and ephemeral integer r_x) are stored in each node for subsequent communications. In case of capturing a node, the intruder will have the ability to derive the private key of the captured node and consequentially he/she can impersonate a legitimate communication partner to the whole network. Even node(s) capturing is detectable, still certificate's revocation is a problem [32] even for implicit certification as certificate management is a prohibiting factor [33].

b) Key revocation and rekeying: When it comes to node capturing or reaching a determined phase of node life time, key revocation, and/or rekeying mechanisms should take place. Unfortunately, Sciancalepore *et al.* [6], [7] could not present a clear vision of the key revocation and rekeying processes to the reader.

III. Proposed Authenticated Key Distribution Protocol

According to our illustration in the related work section, we have shown some endeavors trying to achieve an authenticated key agreement for IIoT using public key cryptography via ECQV implicit certificates. Here, we try to present a much more reliable technique in terms of security and efficiency. We are proposing a key distribution system using an authenticated key broadcasting through the CRT and the use of three nested hash functions. In which the GW will broadcast a ciphered message M_i that carries the session shared key between nodes for each i session as shown in Fig. 2. For that reason, we tried to adapt Lamport's scheme [18] by integrating three different one-way hash functions, $h_A(\cdot)$, $h_B(\cdot)$, and $h_C(\cdot)$ [34], [35].

A. Phase of Key Preloading

IIoT nodes n_j are loaded with three unique CRT modules $r_{j,a}$, $r_{j,b}$, and $r_{j,c}$ that are primes to each other for the whole IIoT network. Furthermore, all device sensors are equipped

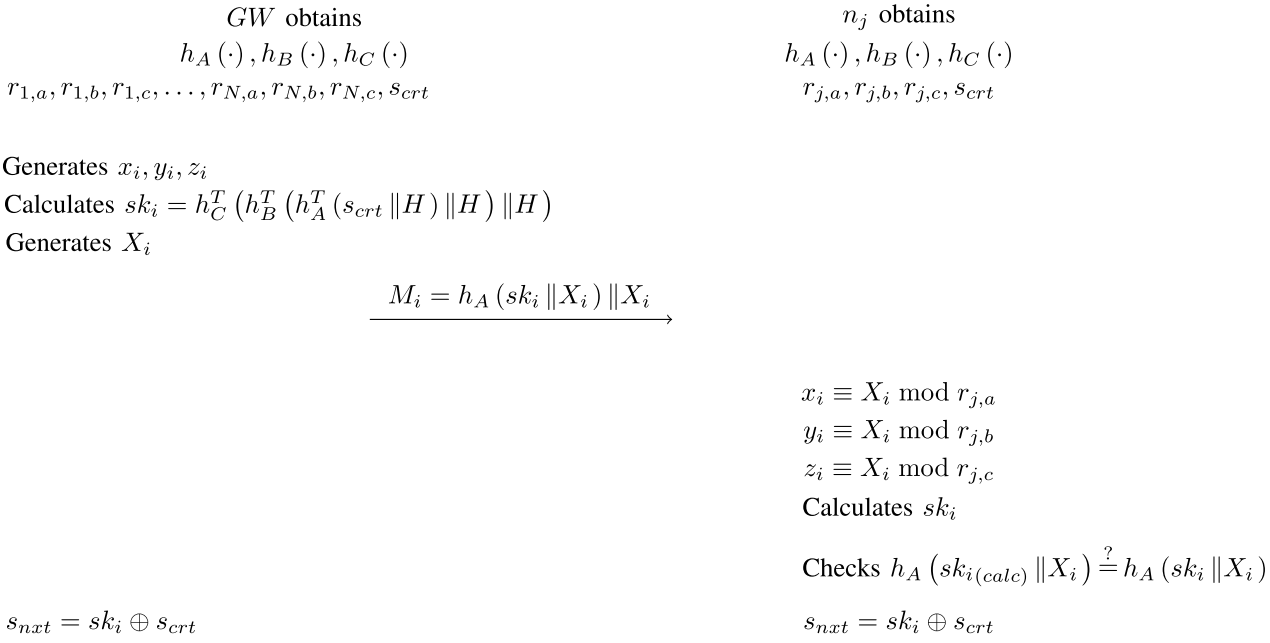

Fig. 3. Proposed key distribution protocol for the IIoT networks.

with a key seed s and the two dissimilar hashes $h_A(\cdot)$ and $h_B(\cdot)$. On the other hand, GW is having the modules of the CRT for all the networks nodes, the key derivation seed s_{crt}, as well as the three dissimilar hash chains.

B. Key Distribution Phase

GW generates three ephemeral random numbers x_i, y_i, and z_i for the ith authentication round, then it calculates the session key

$$sk_i = h_A^T\big(h_B^T\big(h_A^T(s_{crt}\|H)\|H\big)\big\|H\big) \tag{16}$$

where

$$H = x_i\|y_i\|z_i$$
$$T = x_i + y_i + z_i.$$

Accordingly, it calculates the broadcasted chain indices, X_i, for all N nodes using the CRT as follows:

$$\begin{aligned}
X_i &\equiv x_i \bmod r_{1,a}\\
X_i &\equiv y_i \bmod r_{1,b}\\
X_i &\equiv z_i \bmod r_{1,c}\\
X_i &\equiv x_i \bmod r_{2,a}\\
X_i &\equiv y_i \bmod r_{2,b}\\
X_i &\equiv z_i \bmod r_{2,c}\\
&\vdots\\
X_i &\equiv x_i \bmod r_{j,a}\\
X_i &\equiv y_i \bmod r_{j,b}\\
X_i &\equiv z_i \bmod r_{j,c}\\
&\vdots\\
X_i &\equiv x_i \bmod r_{N,a}\\
X_i &\equiv y_i \bmod r_{N,b}\\
X_i &\equiv z_i \bmod r_{N,c}.
\end{aligned}$$

The main purpose of using the CRT is to convey secret key parameters x_i, y_i, and z_i for each node over a one-message communication. It also, the CRT, assures the integrity of x_i, y_i, and z_i. For an intruder to falsified the network nodes with a fake he/she has to have $r_{j,a}$, $r_{j,b}$, and $r_{j,c}$ for the whole network nodes, which is an unrealistic assumption.

GW hashes the session key sk_i concatenated with X_i using a one-way hash function $h_A(\cdot)$ to get $h_A(sk_i\|X_i)$ and then concatenates it once more with X_i to get

$$M_i = h_A(sk_i\|X_i)\|X_i\,. \tag{17}$$

Afterward, the GW broadcasts M_i to all nodes as shown in Fig. 3.

It worth to mention that encrypting the session key with itself, as proposed by [34], can lead to some weaknesses to the session key in terms of key-dependent messages security [36], [37].

C. Key Reception and Verification Phase

Getting M_i by all IIoT nodes helps each node n_j to extract X_i from the concatenation to drive the attached indices x_i, y_i, and z_i using these formulas

$$x_i \equiv X \bmod r_{j,a} \tag{18}$$
$$y_i \equiv X \bmod r_{j,b} \tag{19}$$
$$z_i \equiv X \bmod r_{j,c}. \tag{20}$$

Using the indices x_i, y_i, and z_i, the sensor node n_j will be able to calculate the session key sk_i. A positive comparison between the two hash values, $h_A(sk_{i(\text{calc})}\|X_i) \stackrel{?}{=} h_A(sk_i\|X_i)$ (i.e., calculated and received) will confirm the authentication

process. Otherwise, the received broadcasted message M_i has been altered. Subsequently, a completion of one session will lead to a seed updating, IIoT nodes and GW have to update the current seed s_{crt} to the subsequent one s_{nxt}

$$s_{nxt} = sk_i \oplus s_{crt}. \tag{21}$$

IV. Security Analysis

Referring to the desired security features we have illustrated in Section I, we will assess our approach accordingly.

A. Node Capturing Attack

Node capture attack is a unique and challenging attack for IoT networks. Involving tamper-resistant nodes is not a reliable solution as it increases the network cost extremely [11]. In our protocol, capturing a node for the ith session, will lead to compromise each and everything in the node $h_A(\cdot), h_B(\cdot), h_C(\cdot), r_{j,a}, r_{j,b}, r_{j,c}, s_{crt}$, which in its turn, will result in exploiting sk_i.

In our protocol, we prevent the consequences of capturing a node by revoking the compromised node(s) once it happens and broadcasting new session key parameters minus the CRT modules of the compromised node(s).

This technique works fine provided that nodes' capturing is detectable by the GW. Previous works have shown this assumption to be acceptable in practice [10]–[12]. Therefore, we can say that our protocol is robust to node capture attacks in an implicit way or, in other sense, it can prevent the consequences of these attacks, as the captured nodes will be useless for the intruder.

In case that the GW has detected a capturing attack, it (i.e., the GW) builds and broadcast a new simultaneous congruence X_{i+1} that excludes the compromised node from the network. The new simultaneous congruence X_{i+1} is without the compromised modules $r_{j,a}$, $r_{j,b}$, and $r_{j,c}$ of the compromised node n_j. X_{i+1} will convey a new shared key sk_{i+1} for the rest of nodes to communicate securely, the disclosed information will not be helpful for the attacker. Also s_{crt} will be updated to a new seed value s_{nxt}. In that manner, data confidentiality is guaranteed as illustrated in this case scenario.

For an attacker, who has captured even the next seed s_{nxt}, he/she can run the brute force attack over $sk_i = h_A^T(h_B^T(h_A^T(s_{crt}\|H)\|H)\|H)$ by counting every x_i, y_i, and z_i starting from $x_i = y_i = z_i = 1$, onward, and checking $h_A(sk_i'\|X_i)$ against $h_A(sk_i\|X_i)$, *where* sk_i' *is the cryptanalyzed key*, with a computation complexity of

$$\left(\sum_{z_i=1}^{I}\sum_{y_i=1}^{I}\sum_{x_i=1}^{I}(x_i + y_i + z_i)\right)^3. \tag{22}$$

Taking $I = 7$, for example, this will cost the attacker at least 1.5 years to lunch a successful brute force attack. Considering an attacker platform with an Intel Core i7-6600U CPU @ 2.6 GHz processor and 32 GB RAM. However, in our design we do rekeying every month.

B. Rekeying and Key Revocation

When it comes to reaching a determined key lifetime, *we set it as a one month*, during the ith session, a key revocation and/or rekeying mechanism needs to be considered. On a similar scenario, the GW can initiate the rekeying process by broadcasting a new simultaneous congruence X_{i+1} that carries a new shared key sk_{i+1} for the network nodes. However, this is considered as a revocation if one or more CRT modules are omitted from the new congruence X_{i+1}.

C. Preplay Attack

In our protocol, the GW challenges each sensor node n_j with uniformly distributed and unpredictable values of x_i, y_i and z_i. If we proceed on allowing x_i, y_i, and z_i to take one value of the forward m values, then $p_r(x_i) = p_r(y_i) = p_r(z_i) = 1/m$. Therefore, the successful challenge guessing probability, will be the joint probability of x_i, y_i, and z_i, which is $p_r(x_i, y_i, z_i) = p_r(x_i).p_r(y_i).p_r(z_i) = 1/m^3$. In our implementation we constrained x_i, y_i, and z_i take values from 1 to 7, that reflects $p_r(x_i, y_i, z_i) < 0.003$. We can denote this asset as the capacity of resisting predictable attacks. The restriction of transferring credentials information in unicast distribution, from GW to n_j, increases the robustness against preplay attack.

D. Key Freshness

In our approach, the GW challenges the sensors with random values of x_i, y_i, and z_i. Along with these random values and because of the repeated seed updating, a refreshed keying materials are guaranteed per each session. Accordingly, past session keys cannot be replayed.

E. Server Impersonation

An intruder could impersonate the GW to send an imitated M_i' to the network nodes if he/she can extract the CRT modules $r_{j,a}$, $r_{j,b}$, and $r_{j,c}$ for the whole network, in addition to his/her knowledge about the current seed value s_{crt}, which is a very challenging issue for him/her.

F. Forward Secrecy

Also known as perfect forward secrecy, is a desirable security feature in which the compromisation of session key(s) cannot disclose past session keys. The compromised key(s) and/or current seed s_{crt} are not useful for the attacker to calculate past session keys as he will be facing the intractability of breaking at least $h_C^{z_i}(h_B^{y_i}(h_A^{x_i}(\cdot)))$ to realize previous seeds, *which is needed*, to derive the corresponding session keys.

G. Backward Secrecy

The backward secrecy is a needed security asset in which adding new nodes to the IIoT network must have no influence to disclose or to decrypt any prior information created before their joining, (e.g., *previous session keys*). As random and ephemeral challenges x_i, y_i, and z_i are freshly produced per each session for the network, a result of backward secrecy is assured.

H. Delay Tolerance

We proposed an instant key distribution, *in terms of*, a one exchanged message. Each broadcasted message $M_i = h_A(sk_i \| X_i) \| X_i$ contains the required credentials to authenticate itself and to carry the session key, regardless of previous or next messages.

I. Data Integrity

An inherent validation for data integrity is presented. Any alterations will affect the received value that will be revealed in the key validation phase. Integrity is guaranteed by the CRT, for an intruder to falsified the network nodes with a fake X_i he/she has to have $r_{j\{a,b,c\}}$ for the whole network, which is an unrealistic assumption.

J. Origin Authentication of Data

Hashing the shared key concatenated with X_i provides a direct origin authentication. No one can create the distributed message $M_i = h_A(sk_i \| X_i) \| X_i$ of sk_i and its corresponding X_i except for the GW itself.

K. Key Confirmation

The true key validation in the distribution process declares that the communication's partners have the correct session key. The comparison between the two session keys sk_i, (i.e., *calculated and received*) has the duty of achieving this key confirmation property.

L. Small Challenge Attack

Using a one-way-hash function to build a hash-chain in the backward mode, boost a new sort of attacks named the small challenge attack [20], in which the intruder can extract the hash-chain initial values. The proposed protocol overcomes this weakness by using three dissimilar nested hashes, $h_A(\cdot)$, $h_B(\cdot)$, and $h_C(\cdot)$, to be operated in the onward mode.

V. Formal Security Proof

The formal security analysis of our proposed authentication scheme is achieved using the security protocol verification tool Scyther [13], [16] and the BAN-logic postulate [38].

A. Scyther Tool: Security Protocols Verification

In this section, we prove the security of our protocol using Scyther [16] which is a formal analysis tool of security protocols under the perfect cryptography postulation. We assume that all cryptographic functions are perfect: the attacker cannot learn anything from a ciphered message without the decryption key. Scyther is used to discover problems that arise from the way the protocol is constructed. Scyther evaluates protocols using specific security claims; most of the analyzed protocols using Scyther are widely used standards [26].

Scyther can provide a graphical user interface to make it easier to validate and investigate the protocol. Moreover, attack illustrations are generated by Scyther whenever an attack is found. The language used to write protocols in Scyther is security protocol description language (SPDL). Scyther shows the correctness of the involved security parameters, such as the secrecy (or Secret) as well as the aliveness (or Alive), week agreement (or Weekagree), noninjective synchronization (or Nisynch) and noninjective agreement (or Niagree) as shown in Fig. 4. Synchronization assures that the transferred messages are ordered precisely as agreed by our protocol. Conversely, agreement only cares about the final stage after a successful execution between the communication's partners irrespective to the intermediate stages. Accordingly, synchronization is stronger than agreement in the typical intruder model, as we can say that, synchronized protocols can face replay, suppress-replay, and preplay attacks, however, agreeing could be not able to do so. Aliveness assures that protocol initiator is alive according to the other party during the protocol execution [39], [40]. Week agreement is assured for the communication partners whenever an initiator concludes a run of the protocol, *seemingly with the responder*, then the responder has previously been running the protocol with the initiator [39], [40]. The lack of weak agreement can lead to impersonation and unknown key-share attacks. The full protocol model, its assumptions, the basic security properties, and the algorithm are described in [13]. Table I shows the construction of our protocol SPDL specification.

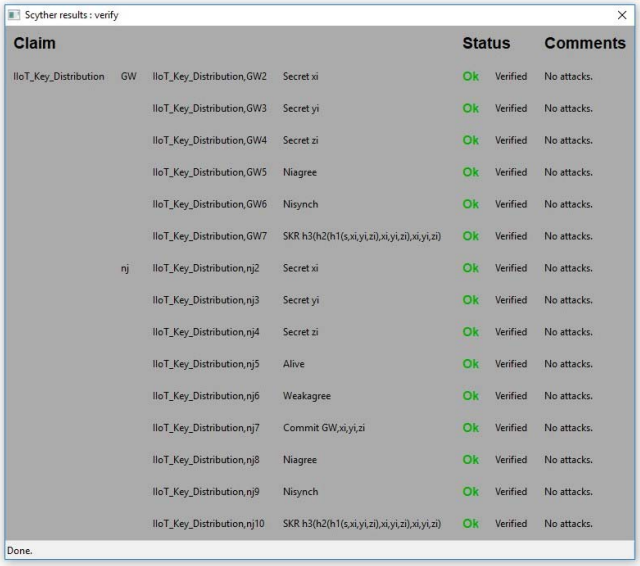

Fig. 4. Validation results of our protocol using Scyther.

Table I shows the construction of our protocol SPDL specification.

Scyther does not support an execution for the CRT modules, i.e., $r_{j\{a,b,c\}}$. Accordingly, we use symmetric encryption to simulate these CRT long-term values [26].

As we present a one way authenticated key distribution, from the GW to n_i, we ignored the claims of Alive, Weakagree, and Commit from the perspective of the GW.

B. BAN Logic Postulate

Through this section, we proof the security of our proposed protocol using the BAN logic postulate [38], [41], [42], as we listed the used hypotheses in Table II.

TABLE I
SPDL SPECIFICATION OF THE PROPOSED PROTOCOL

```
//The Industrial IoT Key Distribution protocol
//The protocol description
//Three different Hash functions
hashfunction h1,h2,h3;
const XOR: Function;
//Seed
secret s;
//The Chinese Remainder Theorem modules
secret rja,rjb,rjc: Function;
protocol IIoT-Key-Distribution (GW,nj)
{
role GW
{
// For the i-th authentication round
fresh xi: Nonce;
fresh yi: Nonce;
fresh zi: Nonce;
macro scrt-GW = s;
//the GW generates the session key for the i-th
//authentication round using three different Hashes
macro ski = h3(h2(h1(scrt-GW,xi,yi,zi),
xi,yi,zi),xi,yi,zi);
//We use symmetric encryption to simulate the CRT
macro Xi={xi}rja,{yi}rjb,{zi}rjc;
//the GW send Mi to sensor nodes
send_1(GW,nj,h1(ski,Xi),{xi}rja(GW,nj),
{yi}rjb(GW,nj),{zi}rjc(GW,nj));
//The GW updates its seed
macro s = XOR(scrt-GW,ski);
claim(GW,Running,nj,xi,yi,zi);
claim(GW,Secret,xi);
claim(GW,Secret,yi);
claim(GW,Secret,zi);
claim(GW,Niagree);
claim(GW,Nisynch);
claim(GW, SKR, ski);
}role nj{
var xi:Nonce;
var yi:Nonce;
var zi:Nonce;
macro scrt-nj = s;
//nj receives Mi and calculates ski'
recv_1 (GW,nj,h1(ski,Xi),{xi}rja(GW,nj),
{yi}rjb(GW,nj),{zi}rjc(GW,nj));
macro ski' = h3(h2(h1(scrt-nj,xi,yi,zi),
xi,yi,zi),xi,yi,zi);
//nj matches the received ski with the calculated ski'
match (h1(ski,Xi), h1(ski',Xi));
//nj updates its seed
macro s = XOR(scrt-nj,ski);
claim(nj,Running,GW,xi,yi,zi);
claim(nj,Secret,xi);
claim(nj,Secret,yi);
claim(nj,Secret,zi);
claim(nj,Alive);
claim(nj,Weakagree);
claim(nj,Commit,GW,xi,yi,zi);
claim(nj,Niagree);
claim(nj,Nisynch);
claim (nj, SKR, ski);}}
```

The message meaning and the nonce verification rules are considered as the fundamental rules of the BAN-logic postulate in which they are defined as follows:

Message Meaning:

$$\frac{P|\equiv P \overset{k}{\leftrightarrow} Q, P \triangleleft \{X\}_k}{P|\equiv Q|\sim X}. \tag{23}$$

Nonce Verification:

$$\frac{P|\equiv \#(X), P|\equiv Q|\sim X}{P|\equiv Q|\equiv X}. \tag{24}$$

TABLE II
BAN LOGIC HYPOTHESES

Construction	Explanation
$P \mid\equiv X$	P believes and trust X
$P \triangleleft X$	P sees X, P got a message that containing X
$P \mid\sim X$	P said X, means that P believed X when he/she sent it
$P \mid\Rightarrow X$	P controls X. P has an authority on X
$\#(X)$	The message X is fresh
$Q \overset{k}{\leftrightarrow} P$	P and Q use K as a shared key
$\{X\}_k$	X ciphered by k
$\overset{k}{\mapsto} P$	k is the P public key
$P \overset{X}{\Leftrightarrow} Q$	The secret formula X is only known to P and Q
$\langle X \rangle_Y$	X combined with a secret Y

C. Our Protocol Idealization

Our protocol is idealized as follows:

$$\mathrm{GW} \rightarrow n_j : h_C^T\Big(h_B^T\Big(h_A^T\Big(\mathrm{GW} \overset{s_{crt},x_i,y_i,z_i}{\longleftrightarrow} n_j\Big)\Big)\Big). \tag{25}$$

D. Initial Assumptions

As we are considering BAN-logic postulate to formally proof our protocol, we need to come with reasonable assumptions for validation, so the assumptions that we relaying on are as follows:

$$n_j|\equiv \#(s_{crt}, x_i, y_i, z_i). \tag{26}$$

Which means that the sensor node n_j believes that the current seed as well as the received challenge is fresh

$$n_j|\equiv n_j \overset{s_{crt},x_i,y_i,z_i}{\longleftrightarrow} \mathrm{GW}. \tag{27}$$

The node n_j believes that the current seed and the random challenge are shared with the GW

$$n_j \triangleleft \Big\{sk_i = h_A^T\Big(h_B^T\Big(h_A^T(s_{crt}\|H)\Big)\|H\Big)\|H\Big)\Big\}. \tag{28}$$

The node n_j sees/receives the shared key sk_i

$$n_j|\equiv \mathrm{GW}|\sim (sk_i). \tag{29}$$

The node n_j believes that sk_i is said by the GW.

E. Anticipated Goals

We need to proof that

$$n_j|\equiv \mathrm{GW}\,|\equiv n_j \overset{sk_i}{\longleftrightarrow} \mathrm{GW}. \tag{30}$$

F. Logic Validation

According to our assumption and by applying the message meaning rule (23) we can get the following:

$$\frac{n_j|\equiv n_j \overset{s_{crt},x_i,y_i,z_i}{\longleftrightarrow} \mathrm{GW}, n_j \triangleleft \{sk_i\}}{n_j|\equiv \mathrm{GW}|\sim (sk_i)}. \tag{31}$$

Then, by applying the nonce-verification rule (24) to (30) we can achieve our authentication goal

$$\frac{n_j|\equiv \#(s_{crt}, x_i, y_i, z_i), n_j|\equiv \mathrm{GW}|\sim (sk_i)}{n_j|\equiv \mathrm{GW}\,|\equiv n_j \overset{sk_i}{\longleftrightarrow} \mathrm{GW}}. \tag{32}$$

Accordingly, we can say that the BAN-logic security proof is validated successfully against our proposed protocol.

TABLE III
COMPARATIVE SUMMARY OF PROPOSED SCHEME WITH ITS RIVALS

Protocol	[8]	[7]	[6]	Proposed
Exchanged Messages	2	4	4	1
Node Operations	$5ECC + MAC$	$ECC + 2HASH + MAC$	$ECC + 2HASH + MAC$	$3\,(x_i + y_i + z_i)\,HASH + HASH + 2MOD$

$HASH$ - Hashing execution time; ECC - Elliptic curve point multiplication time; MOD - Modular arithmetic operation time; MAC - Message authentication code execution time

VI. PERFORMANCE ANALYSIS

Through this section, we will investigate the performance of our proposed protocol in regard to the computational and storage complexity.

A. Storage Examination

For the GW we need to have three keys $r_{j,a}$, $r_{j,b}$, and $r_{j,c}$ per sensor node to build the simultaneous congruence. We also require three dissimilar hashes $h_A(\cdot)$, $h_B(\cdot)$, and $h_C(\cdot)$ as well as one random seed s_{crt}. Therefore, we can express the storage needs as $n.(|r_{j,a}| + |r_{j,b}| + |r_{j,c}|) + |h_A(.)| + |h_B(.)| + |h_C(.)| + |s_{crt}|$, where n is the total number of nodes and $|x|$ is the size occupied in memory by x. To illustrate the memory needs, let us assume $|r_{j,a}| = |r_{j,b}| = |r_{j,c}| = |h_A(.)| = |h_B(.)| = |h_C(.)| = |s_{crt}| = 256$ bits and a network of $n = 1000$ nodes. This will result in a storage requirement of about 32 kb, which is very low by standard storage capabilities of common GW platforms (for example, the several Raspberry Pi models have on-board memory ranging from 256 MB to 1 GB).[2] As for the node storage requirements, we can similarly express them as $|r_{j,a}| + |r_{j,b}| + |r_{j,c}| + |h_A(.)| + |h_B(.)| + |h_C(.)| + |s_{crt}|$, giving us a storage requirement of only 224 bytes under the same assumptions about the size of private keys, random seed, and hashes. Common low-power embedded devices can easily fit this requirement.[3]

B. Computation Examination

In order to compare this paper with previous work [6]–[8], we modeled the most expensive cryptographic operations involved in their execution. The main purpose of this is to obtain a model that allows making an unbiased comparison between the operations required by each scheme in the resource-constrained IoT nodes. For this reason, we ignore the operations performed by the GW in this analysis. With this objective in mind, we also note that the model for our scheme aims to be a tight depiction of the operations involved, while the model for previous works is only lower bound on the operations involved. Our scheme performs the following three modulo operations to compute x_i, y_i, and z_i then it computes $3(x_i + y_i + z_i)$ hashes to obtain $sk_{i|\text{calc}}$. Finally, the node performs an XOR operation to obtain s_{nxt}. The resulting expression is presented in Table III, which also presents the expressions for the operations in previous schemes [6]–[8]. The expressions in Table III were obtained by inspection of [6]–[8]. All three works rely on ECC and, as we will see later in Section IV-C, the cost to execute these operations is significantly higher than the cost to perform hashing on low power embedded nodes. All schemes require precomputed secrets to be presetup on the nodes: $r_{j,\{a,b,c\}}$ and X_i in our scheme and elliptic curve parameters in [6]–[8]. Our model of the computations performed by [6] and [7] only considers the registration phase, in the spirit of modeling a lower bound on the computations. In reality, nodes will have to perform more computations during the secure key establishment phase. Table III also presents the number of messages required by each scheme to establish the keys between two nodes. All nodes receive this message and we model the energy required by each node to receive this message. This contrasts with [6]–[8], where key establishment is always performed between a pair of nodes (but not necessarily between the GW and a node only). The message count was also obtained by inspection of the algorithms and we ignore short protocol messages, such as client/server hello. We also ignore effects of errors that could cause retransmissions, which would be cheaper in our scheme due to the interesting feature of only requiring a single message from the GW.

C. Execution on Embedded Platform

Our scheme only makes use of hash operations and it is expected to have several advantages when compared to other schemes using public key cryptography [6]–[8].

1) Implementation of hashing is simpler (compared to public key cryptography), and it is even possible to find hardware support in low-power embedded platforms.
2) The execution time of our scheme is lower.
3) The overall energy spent by our scheme is also lower.

To support these claims, we started by developing a proof-of-concept implementation of our scheme in the Contiki OS [43],[4] and using publicly available implementations of SHA-224 and SHA-256. Using this implementation, we executed the node-side scheme for different values of x_i, y_i, and z_i such that $x_i = y_i = z_i \wedge x_i, y_i, z_i \in [1, 7] \subset \mathbb{N}$[5] and measured the number of cycles it took in a TI MSP430 micro-controller unit (MCU) using the cycle-accurate emulator MSPSim [44]. Note that this is an MCU used in popular embedded platforms, such as the TelosB, Sky, and Z1. Both the MCU cycles and execution time considering a 8 MHz MCU are presented in Fig. 5. As expected, the execution time increases linearly as we increase x_i, y_i, and z_i because they

[2][Online]. Available: https://www.makershed.com/pages/raspberry-pi-comparison-chart

[3]For example, TI CC26x0 has 20KB SRAM. [Online]. Available: http://www.ti.com/product/CC2650/datasheet

[4][Online]. Available: https://www.dropbox.com/s/hzhqa47jdg7xy2d/key-distr-poc.zip?dl=0

[5]The condition $x_i = y_i = z_i$ is just for implementation propose, however, in the reality x_i, y_i and z_i can take unequal values.

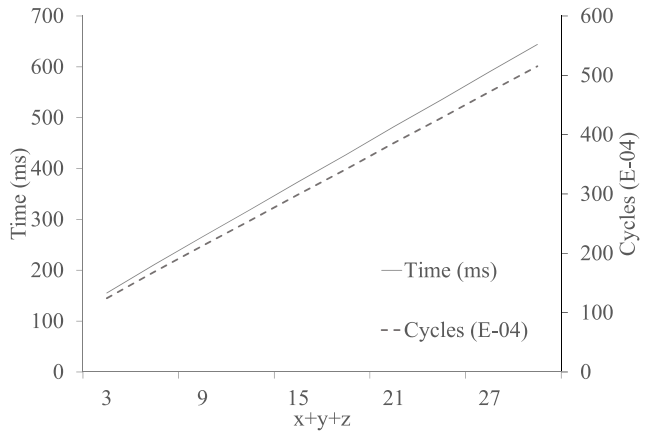

Fig. 5. Execution time and MCU cycles of our proposed scheme as a function of x_i, y_i, and z_i.

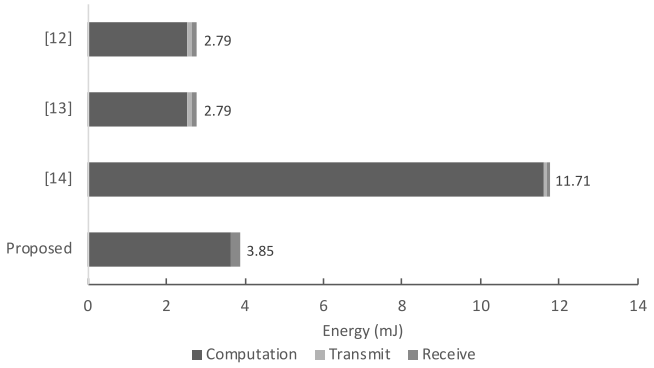

Fig. 6. Computation and communication energy costs (proposed scheme and [6]–[8]).

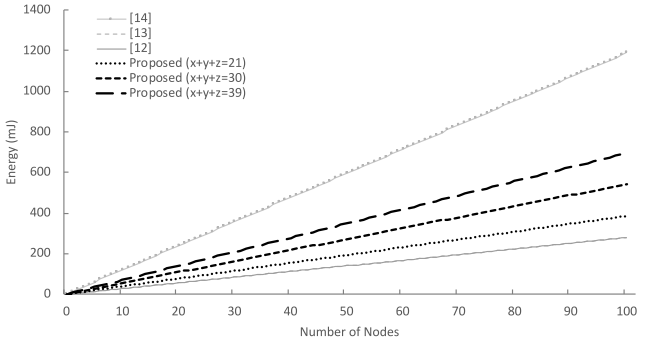

Fig. 7. Computation and communication energy costs as a function of the nodes in the network (proposed scheme and [6]–[8]).

TABLE IV
ENERGY COSTS ON THE TELOSB PLATFORM

Operation	Energy Cost (μJ)
Transmit 1 bit (including header overheads)‡	0.72
Receive 1 bit (including header overheads)‡	0.81
Compute for 1 clock cycle‡	1.20E-03

‡ values from [9].

TABLE V
COMPUTATIONAL AND ENERGY COSTS FOR CRYPTOGRAPHIC OPERATIONS ON THE MSP430 (MCU USED ON THE TELOSB PLATFORM)

Operation	MCU Cycles	Energy Cost (μJ)‡
SHA-224∗	47224	56.67
SHA-256∗	47226	56.67
HMAC-SHA256∗	81522	97.83
64-bit modulo∗	1933	2.32
ECC-159 point mul.⋆	1.92E+06	2.30E+03

∗ measured with cycle-accurate emulator.
⋆ state-of-the-art implementation of fixed-base ECC scalar multiplication on a twisted-Edwards curve [45].
‡ cycles × 1 clock cycle energy.

dictate the amount of hashing operations performed. When our scheme executes $3(x_i + y_i + z_i) = 63$ hashing operations, it takes 482.3 ms. This is an acceptable delay for a substantial challenge to attackers trying to guess the hash-chain (see discussion in Section IV-C) We will now see how the execution time and energy costs of our scheme compares to the previous schemes. For this, we use data from measurements of the execution time of the basic cryptographic operations used in our scheme, the analysis in Section VI-B and the energy costs for processing and communication (for 1 bit of useful payload data, already including header overheads) reported by [9]. Tables IV and V summarize the relevant costs for our analysis, which were measured using a cycle-accurate emulator (to validate the measurements, we checked if they matched the ones reported in [9] for the same operation and MCU). Table V also presents the MCU cycles for ECC operations, using results reported by a state-of-the-art implementation for the same MCU [45]. As expected, Table V shows that ECC operations have a high computational cost in these platforms, when compared to a single hashing operation. We proceed to analyze the resulting energy costs using the energy costs from Table V and the analysis presented in Section VI-B. We consider that our scheme executes $3(x_i + y_i + z_i) = 63$ hashing operations, and will focus on the energy cost of executing the scheme in a single node (ignoring the energy spent by the GW and other nodes). For our scheme, the length of $h_A(sk_i \| X_i)$ is 256 bits, and we assume X_i to be 64 bits of length. Therefore, the total payload of message M_i is 320 bits. For the ECC-based schemes [6]–[8], we assume an average message payload of 89 bits, which is the smallest ECQV M2M certificate size [46]. In practice, not all messages will contain a certificate, but we believe this is a reasonable assumption as messages will often include more information than the certificate and the certificate itself might be bigger (e.g., larger entity size or even longer certificate format, as ECQV M2M is still not widely implemented). As an example, [8] mentions a minimum ECQV certificate size of 193 bits. The energy costs for our scheme compared to the previous schemes [6]–[8] is presented in Fig. 6. We can see that our scheme consumes the same energy for the best previous proposals [6], [7]. We note that we are considering $3(x_i + y_i + z_i) = 63$ hashing operations performed by our scheme, which, as discussed earlier, provides a level of security that is much higher than necessary in practice (see Section IV-C).

Finally, we proceed to study how these energy costs proportions with the nodes number in the network. In this analysis, we consider the computations in the GW as well as the computations and transmit/receive energy costs for all nodes. Other

costs, such as other protocol overheads, idle times, radio listening or acknowledgments and retransmissions are ignored for the purpose of this analysis (as we mentioned previously, retransmissions would be more costly in other schemes). In our scheme, the GW spends energy to transmit a message M_i and all nodes spend energy to receive and compute the shared keys, while other schemes perform message exchange in a pairwise manner. The resulting plot is presented in Fig. 7 and shows that the overall energy expenditure of our scheme grows slower than all previous schemes.

VII. Conclusion

IIoT technology is bringing important innovations to the factory floor. As IIoT deals with vast amounts of privacy-sensitive and security-critical information, it is a primary target for cyberattacks. In this paper, an instant and authenticated key distribution protocol for IIoT has been presented. In the proposed protocol we tried to cover the shortcomings of using implicit certificates in terms of its security weakness to capture attacks as well as its lack of reaching reliable rekeying and key revocation. A detailed security and efficiency analysis show the strength of the proposed protocol against numerous security weaknesses and its advantages compared to state-of-art proposals. We present a formal security proof using the Scyther tool as well as the BAN-logic postulate. We have implemented our scheme in a real platform and evaluated its computing and communication costs under realistic assumptions, using measurements from a cycle-accurate emulator and energy estimations from the previous literature. Our results show that our scheme has energy costs that are competitive when compared to previous proposals, and this energy expenditure scales well when considering the whole network. We note that our scheme has the advantage of relying on simple cryptographic primitives that are easy to implement in hardware for low-power embedded platforms (in fact, it is easy to find off-the-shelf platforms with hardware support for symmetric key encryption and decryption).[6]

References

[1] D. Dzung, M. Naedele, T. P. Von Hoff, and M. Crevatin, "Security for industrial communication systems," *Proc. IEEE*, vol. 93, no. 6, pp. 1152–1177, Jun. 2005.

[2] H. Ning, H. Liu, and L. T. Yang, "Cyberentity security in the Internet of Things," *Computer*, vol. 46, no. 4, pp. 46–53, Apr. 2013.

[3] BusinessWorld. *Value of IoT Estimated at $11 T Yearly in 2025*. Accessed: Jan. 15, 2018. [Online]. Available: http://www.bworldonline.com/content.php?section=Technology&title=value-of-iot-estimated-at-11-t-yearly-in-2025&id=117991

[4] L. Da 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.

[5] A.-R. Sadeghi, C. Wachsmann, and M. Waidner, "Security and privacy challenges in industrial Internet of Things," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, San Francisco, CA, USA, 2015, pp. 1–6.

[6] 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.

[6][Online]. Available: http://www.ti.com/product/CC2650/datasheet

[7] 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.*, Florence, Italy, 2015, pp. 37–42.

[8] 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.*, 2016, pp. 173–179.

[9] G. De Meulenaer, F. Gosset, F.-X. Standaert, and O. Pereira, "On the energy cost of communication and cryptography in wireless sensor networks," in *Proc. IEEE Int. Conf. Wireless Mobile Comput. Netw. Commun. (WIMOB)*, 2008, pp. 580–585.

[10] T. Bonaci, L. Bushnell, and R. Poovendran, "Node capture attacks in wireless sensor networks: A system theoretic approach," in *Proc. 49th IEEE Conf. Decis. Control (CDC)*, Atlanta, GA, USA, 2010, pp. 6765–6772.

[11] S. H. Jokhio, I. A. Jokhio, and A. H. Kemp, "Node capture attack detection and defence in wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 2, no. 3, pp. 161–169, Sep. 2012.

[12] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "Emergent properties: Detection of the node-capture attack in mobile wireless sensor networks," in *Proc. 1st ACM Conf. Wireless Netw. Security*, Alexandria, VA, USA, 2008, pp. 214–219.

[13] C. J. F. Cremers, *Scyther: Semantics and Verification of Security Protocols*. Eindhoven, The Netherlands: Eindhoven Univ. Technol., 2006.

[14] Strategy of Things. *Industrial IoT Versus IoT Do You Know the Difference?* Accessed: May 15, 2018. [Online]. Available: https://strategyofthings.io/industrial-iot

[15] H. Ning, H. Liu, and L. T. Yang, "Aggregated-proof based hierarchical authentication scheme for the Internet of Things," *IEEE Trans. Parallel Distrib. Syst.*, vol. 26, no. 3, pp. 657–667, Mar. 2015.

[16] C. Cremers, *Scyther User Manual*, Dept. Comput. Sci., Univ. Oxford, Oxford, U.K., 2014.

[17] K. Cohn-Gordon, C. Cremers, L. Garratt, J. Millican, and K. Milner, "On ends-to-ends encryption: Asynchronous group messaging with strong security guarantees," IACR Cryptol. ePrint Archive, Rep. 2017/666. 2017. [Online]. Available: http://eprint.iacr.org/2017/666

[18] L. Lamport, "Password authentication with insecure communication," *Commun. ACM*, vol. 24, no. 11, pp. 770–772, 1981.

[19] N. Haller, "The S/KEY one-time password system," Internet Eng. Task Force, Fremont, CA, USA, RFC 1760, 1995.

[20] A. G. Chefranov, "One-time password authentication with infinite hash chains," in *Novel Algorithms and Techniques In Telecommunications, Automation and Industrial Electronics*. Dordrecht, The Netherlands: Springer, 2008, pp. 283–286.

[21] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Proc. IEEE 47th Annu. Allerton Conf. Commun. Control Comput. (Allerton)*, Monticello, IL, USA, 2009, pp. 911–918.

[22] V. Lyubashevsky and D. Masny, "Man-in-the-middle secure authentication schemes from LPN and weak PRFs," in *Advances in Cryptology—CRYPTO 2013*. Heidelberg, Germany: Springer, 2013, pp. 308–325.

[23] M. Campagna, "SEC 4: Elliptic curve qu-vanstone implicit certificate scheme (ECQV)," Certicom Res., Mississauga, ON, Canada, Rep. SEC 4, V. 1.0, 2013.

[24] 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.

[25] 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.

[26] H. Yang, V. A. Oleshchuk, and A. Prinz, "Verifying group authentication protocols by Scyther," *J. Wireless Mobile Netw. Ubiquitous Comput. Depend. Appl.*, vol. 7, no. 2, pp. 3–19, 2016.

[27] A. K. Maurya and V. Sastry, "User authentication scheme for wireless sensor networks and Internet of Things using Chinese remainder theorem," in *Proc. Int. Symp. Security Comput. Commun.*, 2017, pp. 79–94.

[28] J. Zhang, W. Yu, and X. Liu, "CRTBA: Chinese remainder theorem-based broadcast authentication in wireless sensor networks," in *Proc. Int. Symp. Comput. Netw. Multimedia Technol. (CNMT)*, Wuhan, China, 2009, pp. 1–4.

[29] A. Perrig, R. Canetti, J. D. Tygar, and D. Song, "The tesla broadcast authentication protocol," in *Proc. RSA Cryptobytes*, vol. 5, 2005, pp. 2–13.

[30] D. Liu and P. Ning, "Multilevel μTESLA: Broadcast authentication for distributed sensor networks," *ACM Trans. Embedded Comput. Syst.*, vol. 3, no. 4, pp. 800–836, 2004.
[31] E. Barker, D. Johnson, and M. Smid, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)*, document NIST Special Publication 800-56a, Comput. Security, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2007.
[32] P. C. Kocher, "On certificate revocation and validation," in *Proc. Int. Conf. Financ. Cryptography*, 1998, pp. 172–177.
[33] 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. 2017.
[34] M. H. Eldefrawy, M. K. Khan, K. Alghathbar, and E.-S. Cho, "Broadcast authentication for wireless sensor networks using nested hashing and the Chinese remainder theorem," *Sensors*, vol. 10, no. 9, pp. 8683–8695, 2010.
[35] M. H. Eldefrawy and J. F. Al-Muhtadi, "Cryptanalysis and enhancement of a password-based authentication scheme," in *Proc. IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. (CloudCom)*, Vancouver, BC, Canada, 2015, pp. 548–551.
[36] J. Black, P. Rogaway, and T. Shrimpton, "Encryption-scheme security in the presence of key-dependent messages," in *Proc. Int. Workshop Sel. Areas Cryptography*, 2002, pp. 62–75.
[37] D. Hofheinz and D. Unruh, "Towards key-dependent message security in the standard model," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2008, pp. 108–126.
[38] M. Burrows, M. Abadi, and R. M. Needham, "A logic of authentication," *Proc. Roy. Soc. London A Math. Phys. Eng. Sci.*, vol. 426, no. 1871, pp. 233–271, 1989.
[39] G. Lowe, "A hierarchy of authentication specifications," in *Proc. IEEE 10th Comput. Security Found. Workshop*, 1997, pp. 31–43.
[40] C. J. F. Cremers, S. Mauw, and E. P. de Vink, "Injective synchronisation: An extension of the authentication hierarchy," *Theor. Comput. Sci.*, vol. 367, nos. 1–2, pp. 139–161, 2006.
[41] M. S. Alkatheiri, M. H. Eldefrawy, and M. K. Khan, "BAN logic-based security proof for mobile OTP authentication scheme," in *Future Information Technology, Application, and Service*. Dordrecht, The Netherlands: Springer, 2012, pp. 53–59.
[42] P. Syverson and I. Cervesato, "The logic of authentication protocols," in *International School on Foundations of Security Analysis and Design*. Heidelberg, Germany: Springer, 2000, pp. 63–137.
[43] A. Dunkels, B. Gronvall, and T. Voigt, "Contiki—A lightweight and flexible operating system for tiny networked sensors," in *Proc. 29th Annu. IEEE Int. Conf. Local Comput. Netw.*, Tampa, FL, USA, 2004, pp. 455–462.
[44] J. Eriksson, A. Dunkels, N. Finne, F. Osterlind, and T. Voigt, "MSPsim—An extensible simulator for msp430-equipped sensor boards," in *Proc. Eur. Conf. Wireless Sensor Netw. (EWSN) Poster/Demo Session*, vol. 118, 2007, pp. 41–42.
[45] Z. Liu, J. Großschädl, L. Li, and Q. Xu, "Energy-efficient elliptic curve cryptography for MSP430-based wireless sensor nodes," in *Proc. Aust. Conf. Inf. Security Privacy*, 2016, pp. 94–112.
[46] W. Ford and Y. Poeluev, *The Machine-to-Machine (M2M) Public Key Certificate Format draft-ford-m2mcertificate-00.txt*, Internet Eng. Task Force, Fremont, CA, USA, Mar. 2015, pp. 1–14. [Online]. Available: https://tools.ietf.org/html/draft-ford-m2mcertificate-00

Mohamed H. Eldefrawy received the B.Sc., M.Sc., and Ph.D. degrees from the Electrical Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt.

He has been a Senior Researcher with the Center of Excellence in Information Assurance, King Saud University, Riyadh, Saudi Arabia, for almost seven years. He is currently a Post-Doctoral Researcher with the Department of Information Systems and Technology, Mid Sweden University, Sundsvall, Sweden. He has authored or co-authored over 20 research papers in highly reputed journals and conferences. Furthermore, he has invented two U.S./PCT patents in cyber security. His current research interests include wireless network security, digital authentication, and information assurance.

Dr. Eldefrawy was a recipient of the Bronze Medal of the 41st International Exhibition of Inventions, Geneva, Switzerland, in 2013 for one of his inventions. He is a Technical Reviewer for many international journals and conferences.

Nuno Pereira received the Ph.D. degree in computer science from the University of Minho, Braga, Portugal, in 2010.

He is a Professor with the School of Engineering, Polytechnic of Porto (ISEP), Porto, Portugal, He was involved in numerous international (European) and national research projects, including as a Principal Investigator. His research has been published in over 40 technical papers in peer-reviewed scientific venues.

Dr. Pereira regularly serves as a Reviewer and a Technical Committee member for scientific journals and conferences.

Mikael Gidlund (S'98–A'04–M'08–SM'17) received the Lic.Eng. degree in radio communication systems from the Royal Institute of Technology, Stockholm, Sweden, in 2004, and the Ph.D. degree in electrical engineering from Mid Sweden University, Sundsvall, Sweden, in 2005.

From 2008 to 2015, he was a Senior Principal Scientist and a Global Research Area Coordinator of wireless technologies with ABB Corporate Research, Västerås, Sweden. From 2007 to 2008, he was a Project Manager and a Senior Specialist with Nera Networks AS, Bergen, Norway. From 2006 to 2007, he was a Research Engineer and a Project Manager with Acreo AB, Kista, Sweden. Since 2015, he has been a Professor of computer engineering with Mid Sweden University. He holds over 20 patents (granted and pending applications) in the area of wireless communication.