Group Key Establishment for Enabling Secure Multicast Communication in Wireless Sensor Networks Deployed for IoT Applications

PAWANI PORAMBAGE[1], AN BRAEKEN[2], CORINNA SCHMITT[3], ANDREI GURTOV[4], (Senior Member, IEEE), MIKA YLIANTTILA[1], (Senior Member, IEEE), AND BURKHARD STILLER[3]

[1]Centre for Wireless Communications, University of Oulu, Oulu 90014, Finland
[2]Department of Industrial Engineering, Vrije Universiteit Brussel, Brussels 1000, Belgium
[3]Communication Systems Group, Department of Informatics, University of Zürich, Zurich CH-8050, Switzerland
[4]Department of Computer Science and Helsinki Institute for Information Technology, Aalto University, Aalto 00076, Finland

Corresponding author: P. Porambage (pporamba@ee.oulu.fi)

ABSTRACT Wireless sensor networks (WSNs) are a prominent fundamental technology of the Internet of Things (IoTs). Rather than device-to-device communications, group communications in the form of broadcasting and multicasting incur efficient message deliveries among resource-constrained sensor nodes in the IoT-enabled WSNs. Secure and efficient key management is in many cases used to protect the authenticity, integrity, and confidentiality of multicast messages. This paper develops two group key establishment protocols for secure multicast communications among the resource-constrained devices in IoT. Major deployment conditions and requirements of each protocol are described in terms of the specific IoT application scenarios. Furthermore, the applicability of the two protocols is analyzed and justified by a comprehensive analysis of the performance, scalability, and security of the protocols proposed.

INDEX TERMS Internet of Things, wireless sensor networks, multicast, security, group key establishment.

I. INTRODUCTION

The Internet of Things (IoT) has become a powerful element of next generation networking technologies. In an IoT-enabled environment, things or physical objects no longer stay unresponsive. Instead they are connected to the Internet and embedded with processing and communication capabilities. Wireless Sensor Networks (WSNs) determine a key building block of IoT technologies. Typically, sensors are considered resource-constrained devices with limited battery power and computation capabilities (e.g., low CPU clock and memory footprints) [1]. Therefore, it is more effective and efficient to convey multicast messages to a group of devices rather than sending energy consuming unicast messages to individual devices in multiple copies. Securing the group key establishment incline to form the key functionality to provide integrity, authentication, and confidentiality for message transmissions in these multicast groups [2]. Besides, group key establishment protocols have to support device and network characteristics in IoT-enabled WSNs such as resource constraints, scalability, and dynamic group formation.

The field of applying multicast is as manifold as the application area of IoT itself, including smart homes, smart cities, environmental monitoring, and healthcare. For a better understanding of major requirements for a multicast support the following two use cases are determined. The first use case is designed for the control of light bulbs in a smart building [3] (Figure 1a). The environmental monitoring network collects data about light intensity, temperature, and population of all rooms in the building and delivers aggregated data to a central entity. Based on data received, the central entity can enable synchronous operations (e.g., giving commands for on, off, or dim-level) among a group of light bulbs in a floor or room to reach a visual synchronicity of light effects on the user. The second use case is about the collection and aggregation of patient data and sending out the information
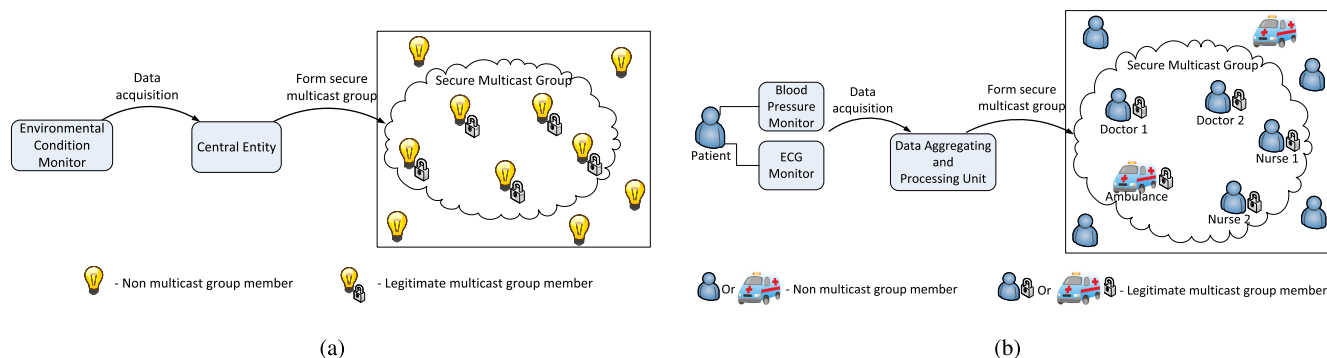
FIGURE 1. Examples of use cases for multicast group creation. (a) Multicasting for light bulbs. (b) Multicasting for medical application.

required to relevant contacts (e.g., doctors or nurses) (Figure 1b). The aggregating unit collects data about the patient's ECG readings and blood pressure. In turn, the processing unit determines the exact set of participants, who should react according to the data acquired, and defines them as a unique multicast group. In these two use cases, multicast groups must be securely formed and respective secret keys have to be shared among all multicast group members to ensure secure communications.

This paper provides the formal modeling of two suitable group key establishment protocols for secure multicasting in IoT-enabled WSN application paradigms. These two protocols are based on Elliptic Curve Cryptographic (ECC) operations. The applicability of these protocols is described in the light of IoT characteristics along with a performance, scalability, and security analysis compared with related work. It is justified that these solutions proposed mitigate the existing security vulnerabilities of those solutions given in the state-of-the-art with better performance characteristics. Moreover, we show a new man-in-the-middle attack on the schemes of [4] and [5].

The remainder of the paper is organized as follows: Section II provides a brief overview of related work. Section III describes the system model, the use case-based adversary model, key assumptions, and the identity-based signature scheme. Section IV discusses the two variants of proposed key establishment protocols in detail. Section V presents assessments of those protocols. Finally, Section VI summarizes the work and draws the conclusions.

II. RELATED WORK
In delivering common messages to a certain group of devices, it is more effective to send multicast messages rather than unicast messages. Multicast communication is recommended for constrained IoT networks to reduce the bandwidth usage, and minimize the energy consumption and processing overhead at the terminals [1]. Establishing a group key among the legitimate members, would enable the secure and trustworthy delivery of messages within a multicast group. Although Datagram Transport Layer Security (DTLS) handshake is designed for device-to-device authentication [6] in IoT,

it does not support multicast security [1]. Security and key management in WSNs is a widely discussed topic [7]–[9].

The WSN group key management protocols such as MIKEY [10] and TESLA [11] are still lacking the compatibility with IoT characteristics. For instance, the MIKEY architecture is entirely designed to facilitate multimedia distributions, whereas TESLA is proposed for the broadcast authentication of the source and not for protecting the confidentiality of multicast messages. Likewise, the Topological Key Hierarchy (TKH) lowers the communication cost of rekeying messages by generating a key-tree based on the underlying topology of WSNs [12]. However, in TKH, the computation and communication costs grow linearly with the number of group members.

Secret sharing is used for different security protocols of WSNs including key management and data confidentiality [4], [13], [14]. The authenticated group key transfer protocol proposed in [4] requires an on-line key generation center (KGC) to construct and distribute the group key, which increases the overhead to implement the system, and reduces flexibility. This work has paved the way to reproduce the keying scheme in [13], which is more dynamic without a trusted KGC. The group key initiator is amongst the group members and all the members equally participate in the final key derivation. However, both schemes [4] and [13] contain pairing-based computations, which do not provide pervasive cipher suites for globally connected IoT devices. Similarly, there are some security vulnerabilities with these schemes as demonstrated in reference [5] such as the uncertainty of tracking the random values of each group member, and the vulnerability to man-in-the-middle (MITM) attacks. We show in Section V that even the solution presented in [5] is not sufficient to resist MITMs.

ECC is a lightweight public key cryptographic (PKC) solution which is defined with standard curve parameters and suitable for securing constrained devices [15], [16]. For instance, [8] and [17] exploit ECC-based implicit certificates and Elliptic Curve Diffie-Helmann (ECDH) algorithm for the secure key establishment in unicast communication in WSNs and IoT. In fact the protocol 1 in Section IV-A is an ECC

variant of reference [13] with improvements (e.g., ensure the integrity and the authenticity of data, and remove the MITM attacks). Protocol 2 (i.e., in Section IV-B) is a further optimized variant of the solution in [4], [5], and [13], and an influenced variant of Elliptic Curve Integrated Encryption Scheme (ECIES). ECIES is a hybrid encryption scheme that uses the functions such as key agreement, key derivation, encryption, message authentication, and hash value computation [18]. Protocol 2 exploits the simplified functionalities in ECIES.

III. SCENARIO AND SIGNATURE SCHEME

This section provides the definitions of the network system and adversary models, and the preliminaries of the identity-based signature scheme used.

A. NETWORK SYSTEM MODEL

The term multicast group stands for a particular group of nodes, which are interested in or entitled to receiving the common set of information or instructions. The total number of nodes considered in the multicast network is n, which includes the initiator node and $(n - 1)$ multicast group members. In the following multicast group members, also known as the responder nodes, are named as U_j for $j = 1, 2, \ldots, (n - 1)$. A common secret key, which is known by the initiator and the responders, is used for secure communication within the multicast group. The key derivation is originated by the initiator and computed according to the inputs given by the responders. For this type of scenario, the size of the multicast network should be equal or greater than four: $n \geq 4$. Otherwise, it would be more efficient when the initiator node derives the group key and delivers the key as unicast messages to both nodes.

B. ADVERSARY MODEL

For the sake of clarity, the behavior of the adversary model is described correlating to the use case of controlling the lights control scenario. According to this example, an adversary can eavesdrop the controlling messages exchanged between the central entity and the light bulbs. It may fraudulently act as a legitimate intermediate device during the key establishment between the central entity and the light bulbs, and launch MITM attacks. Alternatively, an adversary who is external or internal to the network may retransmit the previous key establishment messages to generate replay attacks and interrupt the normal operations of the light bulbs. If the adversary captures a light bulb, he may uncover the secret group keys stored in the bulb.

C. ASSUMPTIONS

Primarily, it is assumed that the underlying communication technology and sensor nodes support multicast group formation and message transactions. Secondly, it is considered that all network entities possess common security associations (i.e., cipher suites) and perform identical cryptographic operations (e.g., hashing ($h()$), encoding, decoding).

Common Elliptic Curve (EC) parameters are embedded in all the network entities that participate in the communication scenario. EC parameters are denoted by q, a, b, G, and p. The parameter q is a prime, which indicates the finite field F_q. The variables a and b are coefficients of EC $y^2 = x^3 + ax + b$, where $4a^3 + 27b^2 \neq 0$. G is the base point generator with order of p, which is also a prime [19]. The initiator (I) is considered a main powered resource rich entity (e.g., gateway node) and has higher processing power and memory capacity than the rest of the nodes in the multicast group. The initiator is also aware of the constitution of the group (i.e., knowing the identities of the legitimate nodes). In both protocols, the initiator is supposed to know the public keys of all the nodes and vise versa. The sleeping patterns of the nodes and path losses in the communication links are not being considered since they are out of the scope of the key objective of this paper. Therefore, it is assumed that the members of the multicast group will eventually receive the initiator requests and the rest of the messages without failures.

D. SIGNATURE SCHEME

By incorporating signatures with the transmitting messages, they would ensure the properties such as integrity, authentication, and non-repudiation. Since the universal accessability of IoT networks are obtained by IPv6 addresses, it would be an added advantage to exploit the device identities with the signature scheme. However, the standard Elliptic Curve Digital Signature Algorithm (ECDSA) does not produce signatures with the node identities. The ECDSA scheme utilizes only the private and public keys of the signee to produce and verify the signature. Therefore, in order to exploit device identities, the following efficient signature scheme is used [20], [21].

Preliminaries: First, the message originator (U_i) selects a random number $r \in \mathbb{Z}_p^*$ and computes $R = rG$. Then the value s is calculated using the originator's identity U_i and the private key d_i, and the hash function h: $s = r + d_i h(U_i||R)$. The function h is a one-way cryptographic hash function that can be deployed in sensor nodes (e.g., SHA-2, SHA-3).

Sign the message:
1) Choose a random number $y \in \mathbb{Z}_p^*$, and compute $Y = yG$.
2) Compute $x = h(U_i||M||R||Y)$ and $z = y + sx$ where M is the message.
3) The signature is (R, x, z).

The signee sends message M along with the signature (R, x, z).

Verify the message:
1) Compute $c = h(U_i||R)$.
2) Check whether $x \overset{?}{=} h(U_i||M||R||(zG - x(R + cQ_i)))$, where Q_i is the public key of the signee (i.e., sender) which is known by the receiver.

IV. PROTOCOL SOLUTIONS DEVELOPED

This section describes the proposed group key establishment protocols. Protocol 1 is a correspondent of the scheme in [13]
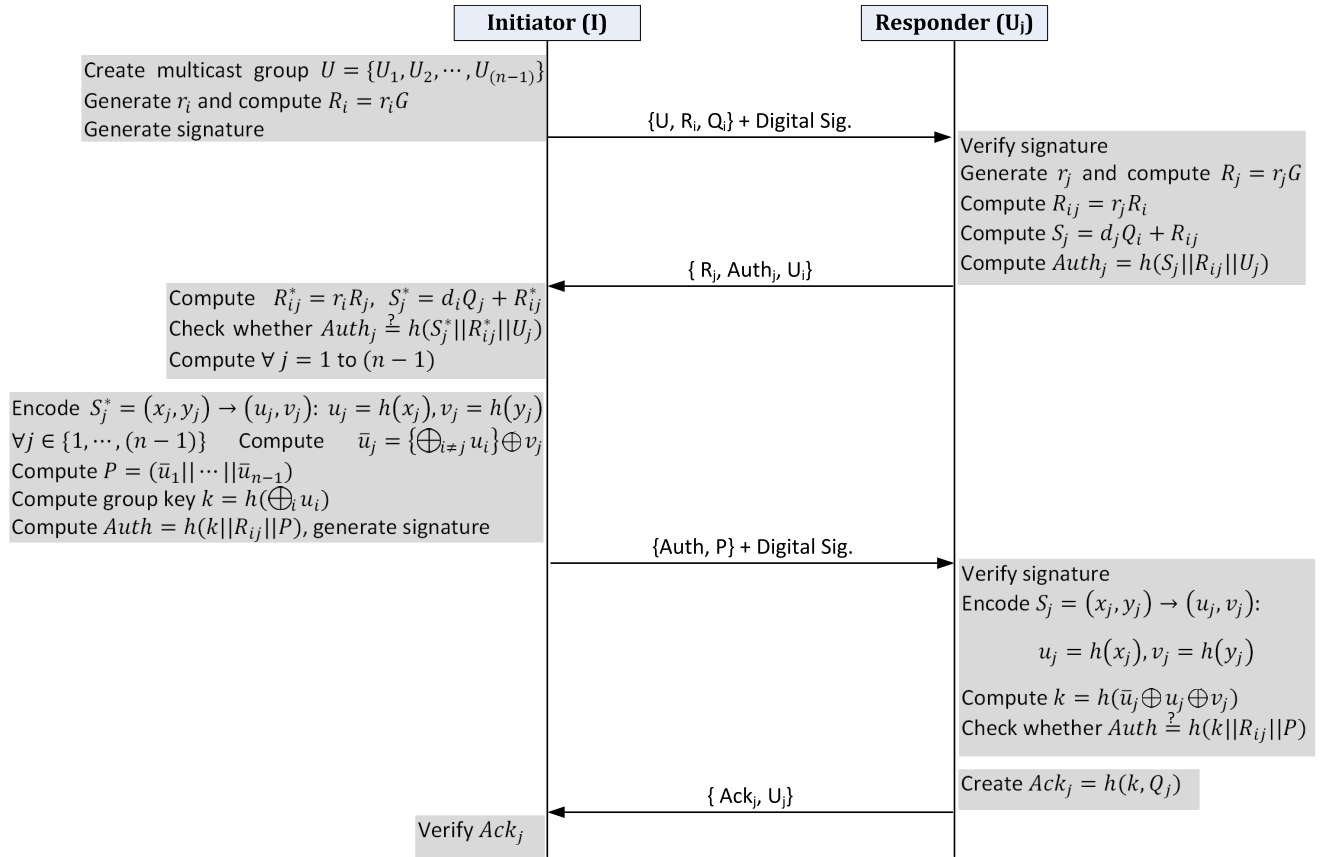
FIGURE 2. Message flow of protocol 1.

after eliminating MITM attacks. Protocol 2 is a more efficient version with the concepts of ECIES.

A. PROTOCOL 1

The message flow of multicast key establishment of protocol 1 is shown in Figure 2. Although the initiator injects the broadcast messages (i.e. to the entire network) to start the key establishment, only the legitimate members of the multicast group are eligible to continue the rest of the process of key derivation.

Step 1: Initiator I determines the set of sensor nodes by their identity that should be included in the particular multicast group, and starts the communication. Accordingly, first, the size of the multicast network (n), and the list of members in the multicast group $U = \{U_1, U_2, \ldots, U_{(n-1)}\}$ are defined by the initiator. Then a random number $r_i \in \mathbb{Z}_p^*$ is generated for the particular multicast session in order to obtain the freshness of each session and $R_i = r_i G$ is computed. The broadcasting message is created using I's public key $Q_i = d_i G$, R_i, and U. Later, the message $\{Q_i, r_i, U\}$ is broadcast to the entire network along with the digital signature of the message, in order to announce the initiation of the multicast communication. Digital signature is computed as stated in Section III-D. Parameter R_i in protocol 1 is reused for the parameter R in the signature scheme, whereas parameter Y in the signature scheme should be freshly obtained.

Step 2: When the initial message is received by the sensor nodes in the network, first the list U is checked by each node to verify whether the particular node is included in the multicast group. If the node identity U_j, for $j = 1, 2, \ldots, n - 1$, is included in the list, the message is further processed, else it is discarded. The integrity of the received message is verified from the digital signature value. A freshly generated random number $r_j \in \mathbb{Z}_p^*$ and R_i values are used to compute R_{ij} EC point, $R_{ij} = r_j R_i$. $R_j = r_j G$ is also calculated for using shortly. R_{ij} value, U_j's private key d_j, and initiator's public key Q_i are used to compute the secret EC point S_j: $S_j = d_j Q_i + R_{ij}$. Afterwards, U_j computes $Auth_j = h(S_j \| R_{ij} \| U_j)$, and sends $\{R_j, Q_j, Auth_j, U_j\}$ to the initiator as a response.

Step 3: Initiator I collects the responses received from all the responders $j = 1$ to $(n - 1)$. If there is a loss of responses from the listed nodes in the multicast group, the initiator re-sends the same message after a retransmission time-out. For the retransmission it can use the same sequence number with a different epoch according to the DTLS handshaking mechanism [6]. However, further information about the retransmission is not provided, since it is out of scope of the main goal of the protocol design. After receiving the message from responder U_j, EC point S_j^* is computed by the initiator. The r_j and Q_j values are used from the received message.

$$r_{ij}^* = r_i . r_j \bmod p, \quad R_{ij}^* = r_{ij}^* G, \quad S_j^* = d_i Q_j + R_{ij}^*$$
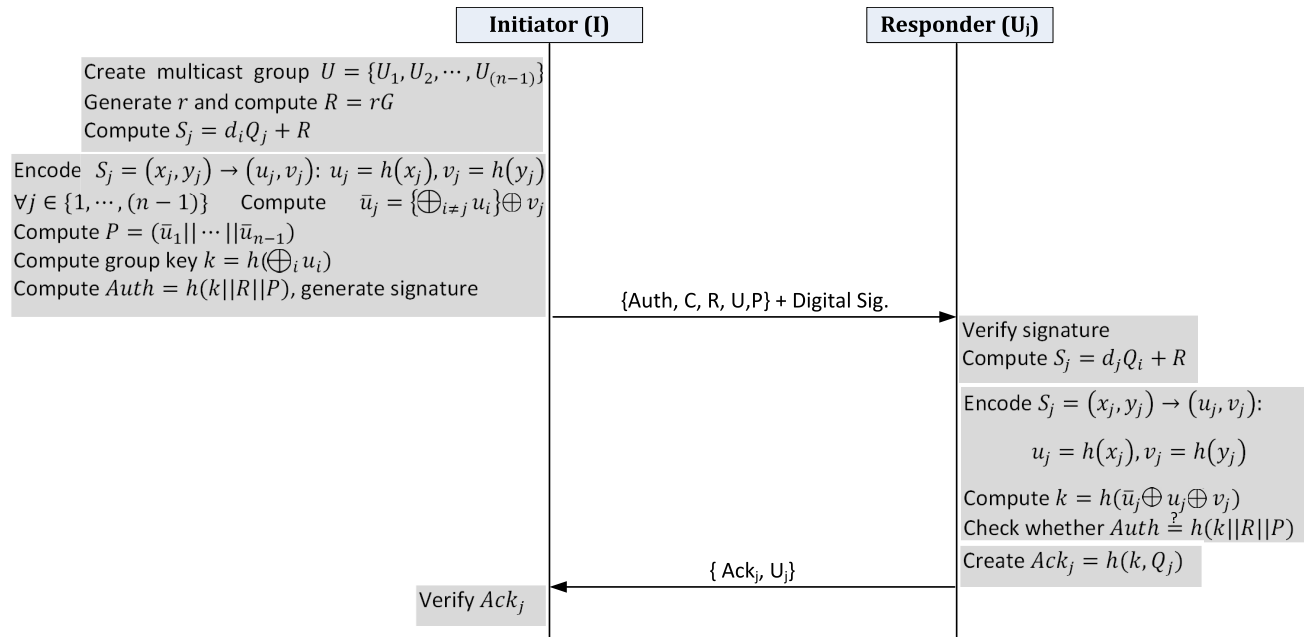
FIGURE 3. Message flow of protocol 2.

Then the initiator checks $Auth_j \stackrel{?}{=} h(S_j^* \| R_{ij}^* \| U_j)$. If the verification is successful, the intiator can proceed to the next step. Otherwise, it discards the message and re-sends the same multicast initiation request to those particular sensor nodes. If the verification result is still not successful for the retransmissions of a certain node, then the initiator discards that node from the multicast group.

Step 4: As aforementioned in step 3, the initiator I computes the respective S_j EC points (i.e., shared secrets) for all the nodes of the multicast group. EC point $S_j = (x_j, y_j)$ is encoded into the point (u_j, v_j) as follows: $u_j = h(x_j)$; $v_j = h(y_j)$. Next, $\forall j \in \{1, \ldots, n-1\}$, the valuee $\bar{u}_j = \{\bigoplus_{i \neq j} u_i\} \oplus v_j$ are computed. The set $P = (\bar{u}_1 \| \ldots \| \bar{u}_{n-1})$ is determined and the multicast group key is then defined as $k = h(\bigoplus_i u_i)$.

The new *Auth* code is now calculated as follows: $Auth = h(k \| R_{ij} \| P)$. Afterwards, the initiator broadcasts the message *Auth, P* along with the digital signature, which is computed as described in Section III-D. The random value R_i is reused as parameter R in the signature scheme.

Step 5: When a responder node U_j receives the second broadcast message, it first verifies the digital signature. The responder U_j uses S_j to compute (u_j, v_j) point. Next, the key k can be derived by $k = h(\bar{u}_j \oplus u_j \oplus v_j)$. Then U_j verifies whether $Auth \stackrel{?}{=} h(k \| R_{ij} \| P)$. If this is correctly verified, then the group key k is authenticated.

Step 6: Each sensor node should send an acknowledgement message $h(k, Q_j)$ to finish the handshake. This ensures that every group member has correctly derived the group key k.

After six steps, the initiator I and the other members of the multicast group U are having a common secret key k that can be used for multicast communication among the group.

B. PROTOCOL 2

Protocol 2 exploits the concepts of ECIES to establish a shared secret key among the multicast group (Figure 3).

Step 1: First, the size (n) and the composition of the multicast group $U = \{U_1, U_2, \ldots, U_{(n-1)}\}$ are determined by the initiator as done in step 1 in protocol 1. Then a random value r is generated, where $R = rG$. EC points S_js are computed using r and the public keys Q_j of the group members: $S_j = d_i Q_j + R$, where $j = 1$ to $n-1$. Similar to protocol 1, EC point $S_j = (x_j, y_j)$ is encoded into the point (u_j, v_j) as follows: $u_j = h(x_j)$; $v_j = h(y_j)$. Similarly, $\forall j \in \{1, \ldots, n-1\}$, the values $\bar{u}_j = \{\bigoplus_{i \neq j} u_i\} \oplus v_j$ are computed and denoted in the set $P = (\bar{u}_1 \| \ldots \| \bar{u}_{n-1})$. The secret key is then defined as $k = h(\bigoplus_i u_i)$.

The *Auth* code is calculated as follows: $Auth = h(k \| R \| P)$. The new multicast message for group U is generated and transmitted by the initiator with the calculated values and the counter value C as follows: $(Auth, C, R, U, P)$. Additionally, the digital signature is appended to preserve message authentication and integrity. The same R value can be reused as the parameter R in the signature scheme in Section III-D.

Step 2: When the sensor node U_j receives the broadcast message, initially, it checks whether it is included in the multicast group U. Then the digital signature and the counter C are checked. If both are correctly verified, S_j is computed using the received random value R and node's private key d_j: $S_j = d_j Q_i + R$. The EC point S_j is converted to the point (u_j, v_j) using the same encoding as in step 1. Next, the key k is derived by $k = h(\bar{u}_j \oplus u_j \oplus v_j)$. Similarly, all the nodes in the group have to proceed the same computations to derive the group key. Then U_j verifies whether $Auth \stackrel{?}{=} h(k \| R \| P)$. If this is correctly verified, then the group key k is authenticated.

TABLE 1. Computational overhead and message length of key establishment protocols.

| | Protocol 1 | | Protocol 2 | |
	Initiator	Responder	Initiator	Responder
Complete computational overhead	$(2n+1)PM+(n-1)PA$	$9PM+5PA$	$(n+1)PM+(n-1)PA$	$4PM+3PA$
Length of sent messages (Byte)	$2n+134$	56	$2n+84$	18
Length of received messages (Byte)	$56(n-1)$	$2n+134$	$18(n-1)$	$2n+84$

Step 3: Each sensor node should send an acknowledgement message $h(k, Q_j)$ to finish the handshake. Later, by verifying the acknowledgement message, the initiator can ensure the authenticity of the particular group member and the accurate derivation of group key k.

After three steps the shared secret key is known by the initiator and the other members in the multicast group. Compared to protocol 1, this protocol 2 is more efficient and creates lower overhead on the sensor nodes due to less message transactions and reduced number of operations at the responder ends.

V. PROTOCOL ANALYSIS

The performance analysis is based on the estimated energy consumption of the computation and communication energy cost of the protocols. The scalability analysis illustrates the protocol behaviors at node additions and removals. Security analysis explains how well the proposed protocols can mitigate the most common security threats and vulnerabilities. We also show a new MITM attack on the schemes of [4] and [5].

A. PERFORMANCE EVALUATION

For the key establishment, the number of message transactions between the initiator and a responder group member is four for protocol 1 and two for protocol 2. Additionally, the number of operations performed at each end, the number of message transactions, and the overhead are also less in protocol 2 than that of protocol 1 as shown in Table 1. This increases the efficiency and performance of the second proposed protocol. However, in both protocols, the group key has to be re-established after the addition of a new node or the removal of an existing node. In both protocols, in order to provide group and initiator authentication, the group key is derived with the contribution of the multicast group members (i.e, the group key is derived by xoring the key components of each member). This is an implicit assurance that all nodes contribute and authorize the final group key. However, in protocol 1 the group members provide greater contribution to the key derivation with a higher degree of randomness, whereas in protocol 2 the initiator performs the majority of the operations.

Comparing to hashing and xoring operations, EC point operations (i.e., point addition and multiplication) are considered the most expensive calculations. Therefore, in order to estimate the approximate energy consumptions for computation, message transmission, and message reception,

we neglect those operations that induce smaller impact on the total energy, and consider only the EC point multiplications (PM) and point additions (PA) in each step. Accordingly, Table 1 provides the computational overhead and the length of transmission and reception messages, when the multicast network size is n. Calculations are performed for the `secp160r1` curve ECC operations with the estimations such as EC point is 20 Byte, $h()$ output is 16 Byte, node identity and counter C are 2 Byte, and value P is 16 Byte. The final values also include the contribution of the digital signature scheme as explained in Sections III-D and IV. Moreover, in the actual implementation it is necessary to perform the fragmentation of the large messages, which exceed the maximum transfer unit size of the network (e.g., in IEEE 802.15.4 networks this would be 128 Byte).

Energy costs are computed with respect to standard Crossbow TelosB sensor nodes, which embed 4 MHz MSP430 microcontroller and comply with the IEEE 802.15.4 standards with a data rate of 250 kbps. According to [22], energy values are approximated taking into account that EC point multiplication consumes 17 mJ and the point addition also has an upper bound of the same value. From the characteristics of the CC2420 transceiver used in TelosB sensors, the unit transmission and reception energy costs are respectively taken as 0.209 μJ and 0.226 μJ. Accordingly the computation, transmission, and reception energy consumptions are calculated for both protocols 1 and 2 at the responder sides by varying the size of the network n along with the TKH scheme [12], as shown in Figure 4. As depicted in the figure,
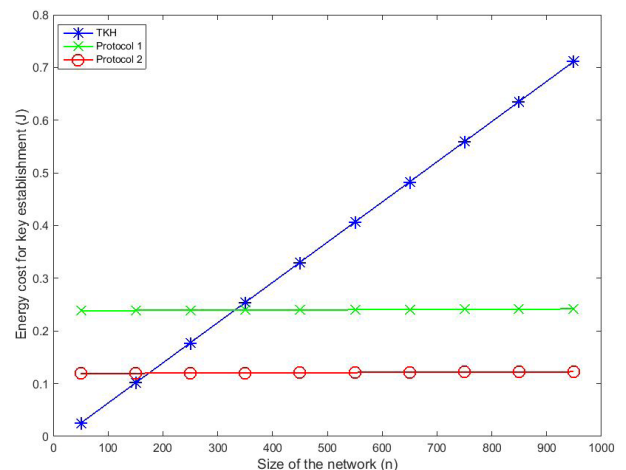

FIGURE 4. Total energy costs for group key establishment protocols.

the energy costs of the key establishment at end nodes in our protocols are reasonably lower than the tree-based TKH scheme for large group sizes.

Furthermore, the complete computational overheads at the responder side for both protocols remain almost constant irrespective of the size of the multicast group. Protocol 2 outperforms protocol 1 with a factor of almost two with respect to computation, a factor of almost three with respect to transmission, and a fixed amount of 11.3 μJ for reception energy.[1] The complete computation energy for protocol 1[2] is approximately 238 mJ and for protocol 2[3] it is 119 mJ. Taking into account that with two Zinc-carbon AA batteries of 1.5 V nominal voltage and 800 mAh average capacity, the available energy[4] is 8640 J. Consequently, these values correspond to 0.0027% of the total available energy for one complete execution of protocol 1, and 0.0017% of that of protocol 2. Taking only the execution of these protocols into account, it implies that protocol 2 (i.e., at the responder side) can execute the key agreement around 57600 times, while protocol 1 can execute half of it.

B. SCALABILITY

For the ease of explanation, protocol 2 is first taken into account for discussing the scalability features as it has less message transactions. The actions are described with respect to the key refreshing when a new member joins or an old member leaves the group. When a new member U_x joins, the initiator node needs to compute $S_x = d_iQ_x + R$. Otherwise a unicast message needs to be sent to U_x. The corresponding EC point (u_x, v_x) is derived from S_x. Next a new random key k is derived. The rest of the protocol remains the same. The difference with key refreshing is that $n - 1$ less point multiplications need to be performed in order to derive the points associated to the group members since those points are pre-calculated. The message length on the other hand slightly increases with one extra value for $\bar{u}_x$ and the length of the identity U_x.

On the other hand when a member U_o leaves the group, the initiator node needs to determine a new group key k, using the $n - 2$ remaining values of u_i. Now the transmission can be simplified, since only an updated version of C, the point R, the removed user U_o, together with an authentication tag, and a signature need to be sent. As a consequence, the message length reduces by $(n - 1) * 20 + (n - 2) * 2$ Byte. This is only valid, if the node stores the information of those points related to the users. Similar adaptations are performed in protocol 1 at node addition and node removal. The significant difference in the node addition in protocol 1 is that message 1 and 2

---

are unicast message exchanges between the initiator and new node U_x. The initiator computes only the new EC point S_x and reuses the remainder of the pre-computed $(n - 1)$ points. When leaving a member in protocol 1, the initiator can reuse the pre-calculated $(n - 2)$ points and determine a new group key k.

C. SECURITY ANALYSIS

The first important security feature of the proposed protocols is the guarantee for integrity and authenticity of the message transactions. This follows from the fact that every transmitted message contains either a digital signature or a hash in which identity-related information is included and can be verified by the intended receiver. The correctness of the data source can be guaranteed by the fact that each node has its own private key, together with the public key of the initiator. The initiator possesses besides its own private key, the list of active nodes together with their corresponding public keys. Furthermore, denial-of-service (DoS) attacks are also mitigated by the exploitation of digital signatures. Evidently in both protocols, an eavesdropper is unable to derive the group key only by analyzing messages transmitted since, the responder derives the final key using the content of the messages received (P) and its own secret value (EC point S_j).

As indicated in the particular adversary model in Section III-B, an adversary may impose MITM attacks during the key establishment process. A new MITM attack model is explained in [5] for the scheme proposed in [4], which is taken as the reference for protocol 1. In our work, MITM attack resistance is provided for protocol 1 by including digital signatures. In an MITM attack it is possible to change the group of intended members in step 1. The key of the solution in [5] consists of an addition of a digital signature to the message containing the group of intended members. However, they do not notice that a possible MITM attack can occur even at the end of the protocol, step 4. By translating this knowledge to the protocols developed here, in step 4 of [5], the initiator broadcasts the following message: $P_1, \ldots, P_{n-1}, \quad h(k\|P\|R_1\|\ldots\|R_{n-1})$. Values R_i's denote random values generated by participating nodes in step 3.

This leads to two problems. First of all from a practical point of view, it is not evident that each node keeps track of individual random values r_j of other nodes. Secondly, a MITM attacker can intercept this message and forward a new message defining a key k that he determines himself and wants to share with $n - 2$ other participants. Suppose the attacker U_a' wants to exclude the intended user U_a from the list of users for any a between 1 and $n - 1$. Therefore, a key will be constructed among the trusted users $U_1, \ldots, U_{a-1}, U_{a+1}, \ldots, U_{n-1}$ together with the attacker U_a'. In order to proceed with this attack, the attacker needs to do two things: First, he intercepts the message of step 3 from U_a: $r_a, Q_a, U_a, h(S_a\|R_{ia}\|U_a)$ and solely forwards this to the initiator. The message $r_a', Q_a, U_a, h(S_a\|R_{ia}\|U_a)$ is sent to the other nodes. They store the random value r_a' associated with user U_a, while it

---

is in fact user U_a'. This becomes possible, since they are not able to check the hash value, because they do not know the private key of the initiator. Secondly, in step 4, the attacker can now reuse $n-2$ values $\bar{u}_1, \ldots, \bar{u}_{a-1}, \bar{u}_{a+1}, \ldots, \bar{u}_{n-1}$ of the original message generated by the initiator and combine it with his own value u_a. He also determines his own secret key k to be shared with the others and forwards the message to the other nodes: $\bar{u}_1, \ldots, \bar{u}_{a-1}, \bar{u}_a', \bar{u}_{a+1}, \ldots, \bar{u}_{n-1}$, and $h(k\|\bar{u}_1\| \ldots \|\bar{u}_{a-1}\|\cdot u_a\|\bar{u} \ldots_{a+1} \| \ldots \|\bar{u}_{n-1} \|R_1\| \ldots \|R_{a-1}\| R_a'\|R_{a+1}\| \ldots \|R_{n-1})$.

To conclude, in order to solve these two problems (i.e., tracking individual's random values and MITM attacks), the removal of R_i is proposed in the hash of the message of step 4. Moreover, it is clear that the addition of a digital signature is required for resistance against MITM attacks. Furthermore, the following security considerations apply to the two new protocols:

- When a sensor node is compromised by an attacker or not needed anymore, it will be removed from the network and also from the list of active sensors, stored in the initiator node. Since each node contains a unique private key, and the established key is dependent on the collaboration of $n-1$ users, cryptographic material from a compromised node cannot be used to determine the key.

- Any cryptographic key has to be updated regularly. For instance, if a node is compromised and not yet detected by the initiator, the attacker can actively join with all communications in the network. However, since the group key is dependent on random values and on the input of the other participants, it is not possible to decrypt previous messages from the past, if less than $n-2$ nodes of the group are compromised. Consequently, key updates (besides the group key, including private and public keys of nodes) have to be performed on a regular base is dependent on the load of the traffic.

- Replay attacks described in the adversary model are made impossible due to the use of random values by each participant. Even if a node uses the same random value r_j as before, the used random parameter R_{ij} will still be random, since it also depends on the random value r_i of the initiator.

As steps 1 to 4 in protocol 1 correspond to step 1 in protocol 2, it can be derived that protocol 2 is also secured.

VI. SUMMARY, CONCLUSIONS, AND FUTURE WORK

This paper designed and analyzed two secure group key establishment mechanisms for multicasting in WSNs in the context of IoT applications. The key derivations also implicitly authenticate group members, whereas the key can be further used for securing multicast messages.

According to the performance evaluations results, computation and communication energy consumptions of both protocols are tolerable by the resource-constrained sensor nodes. The security analysis reassures the stronger security features of those protocols proposed compared to

reference solutions. Scalability properties of these protocols ensure the support of frequent changes of the multicast group. Although scalability and security characteristics are closely coupled with both protocols, protocol 2 always outperforms protocol 1 in terms of energy consumption. Protocol 1 is more appropriate for distributed IoT applications, which require group members to highly contribute to the key computation and need greater randomness. Since the energy cost at the responder side is very low, protocol 2 is more suitable for centralized IoT applications, where mostly cryptographic operations are performed by a central entity and edge nodes have very low energy profiles. The two protocols proposed are applicable to one-to-many $(1 : n)$ communication scenarios and they are expected to be extended to many-to-many $(m : n)$ communication scenarios obtaining comprehensive quantitative results for real-time test-beds.

REFERENCES

[1] S. Keoh, S. Kumar, O. Garcia-Morchon, E. Dijk, and A. Rahman. (Feb. 2014). *DTLS-Based Multicast Security for Low-Power and Lossy Networks (LLNs)*. [Online]. Available: http://tools.ietf.org/pdf/draft-keoh-dice-multicast-security-05

[2] J. Zhang and V. Varadharajan, "Wireless sensor network key management survey and taxonomy," *J. Netw. Comput. Appl.*, vol. 33, no. 2, pp. 63–75, 2010.

[3] Rahman and E. Dijk. (Oct. 2014). *Group Communication for the Constrained Application Protocol (CoAP)*. [Online]. Available: https://tools.ietf.org/html/rfc7390

[4] L. Harn and C. Lin, "Authenticated group key transfer protocol based on secret sharing," *IEEE Trans. Comput.*, vol. 59, no. 6, pp. 842–846, Jun. 2010.

[5] W. Yuan, L. Hu, H. Li, and J. Chu, "Security and improvement of an authenticated group key transfer protocol based on secret sharing," *Appl. Math. Inf. Sci.*, vol. 7, no. 5, pp. 1943–1949, 2013.

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

[7] P. Nie, J. Vähä-Herttua, T. Aura, and A. Gurtov, "Performance analysis of HIP diet exchange for WSN security establishment," in *Proc. 7th ACM Symp. QoS Secur. Wireless Mobile Netw.*, 2011, pp. 51–56.

[8] P. Porambage, P. Kumar, C. Schmitt, A. Gurtov, and M. Ylianttila, "Certificate-based pairwise key establishment protocol for wireless sensor networks," in *Proc. IEEE 16th Int. Conf. Comput. Sci. Eng. (CSE)*, Dec. 2013, pp. 667–674.

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

[10] J. Arkko, E. Carrara, F. Lindholm, M. Naslund, and K. Norrman. (Aug. 2004). *MIKEY: Multimedia Internet Keying*. [Online]. Available: http://www.rfc-base.org/txt/rfc-3830.txt

[11] A. Perrig, D. Song, R. Canetti, J. D. Tygar, and B. Briscoe. (Jun. 2005). *Timed Efficient Stream Loss-Tolerant Authentication (TESLA): Multicast Source Authentication Transform Introduction*. [Online]. Available: http://www.ietf.org/rfc/rfc4082.txt

[12] J.-H. Son, J.-S. Lee, and S.-W. Seo, "Topological key hierarchy for energy-efficient group key management in wireless sensor networks," *Wireless Pers. Commun.*, vol. 52, no. 2, pp. 359–382, 2010.

[13] C.-Y. Lee, Z.-H. Wang, L. Harn, and C.-C. Chang, "Secure key transfer protocol based on secret sharing for group communications," *IEICE Trans. Inf. Syst.*, vol. 94, no. 11, pp. 2069–2076, 2011.

[14] R. Di Pietro and S. Guarino, "Data confidentiality and availability via secret sharing and node mobility in UWSN," in *Proc. IEEE INFOCOM*, Apr. 2013, pp. 205–209.

[15] Certicom Research, Standards for Efficient Cryptography. (Sep. 2000). *SEC 2: Recommended Elliptic Curve Domain Parameters, Version 1.0*. [Online]. Available: http://www.secg.org/SEC2-Ver-1.0.pdf

[16] National Institute of Standards and Technology. (Aug. 1999). *Recommended Elliptic Curves for Federal Government Use*. [Online]. Available: http://csrc.nist.gov/groups/ST/toolkit/documents/dss/NISTReCur.pdf

[17] 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. 2014, Jul. 2014, Art. ID 357430.

[18] V. Shoup. (2001). *A Proposal for an ISO Standard for Public Key Encryption (Version 2.0)*. [Online]. Available: http://www.shoup.net/papers/iso-2_1.pdf

[19] D. Hankerson, A. J. Menezes, and S. Vanstone, *Guide to Elliptic Curve Cryptography*. New York, NY, USA: Springer-Verlag, 2003.

[20] X. Cao, X. Zeng, W. Kou, and L. Hu, "Identity-based anonymous remote authentication for value-added services in mobile networks," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3508–3517, Sep. 2009.

[21] H. Yu, J. He, R. Liu, and D. Ji, "On the security of data collection and transmission from wireless sensor networks in the context of Internet of Things," *Int. J. Distrib. Sensor Netw.*, vol. 2013, Aug. 2013, Art. ID 806505.

[22] G. de Meulenaer, F. Gosset, O.-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.*, Oct. 2008, pp. 580–585.

CORINNA SCHMITT received the Diploma degree in bioinformatics from the University of Tübingen, in 2006, and the Ph.D. degree in 2013. She joined the Department of Computer Science, Technische Universität München, with a focus on wireless sensor networks for her Ph.D. degree in 2008. She established an efficient data transmission protocol called TinyIPFIX with additional features for aggregation, compression, secure transmission, and a user-friendly and flexible Graphical User Interface (CoMaDa). She joined the Communication Systems Group (CSG), Department of Informatics, University of Zürich, Switzerland. She is currently the Head of Mobile and Trusted Communications with CSG.

ANDREI GURTOV (SM'10) received the M.Sc. and Ph.D. degrees in computer science from the University of Helsinki, Finland, in 2000 and 2004, respectively. He was a Professor of Wireless Internet with the University of Oulu from 2010 to 2012. He was with TeliaSonera, the Ericsson Nomadic Laboratory, and the University of Helsinki. He was a Visiting Scholar with the International Computer Science Institute, Berkeley, in 2003, 2005, and 2013. He is currently a Principal Scientist with the Helsinki Institute for Information Technology. He is also an Adjunct Professor with Aalto University, the University of Helsinki, and the University of Oulu. He has co-authored over 150 publications, including three books, research papers, patents, and five IETF RFCs. He is a Senior Member of the Association for Computing Machinery.

PAWANI PORAMBAGE received the B.Sc. degree in electronics and telecommunication engineering from the University of Moratuwa, Sri Lanka, in 2010, and the M.Sc. degree in ubiquitous networking and computer networking from the University of Nice Sophia Anipolis, France, in 2012. She is currently pursuing the Ph.D. degree with the Centre for Wireless Communication (CWC), University of Oulu, Finland. She was a Visiting Researcher with the Communication Systems Group, Department of Informatics, University of Zurich, Switzerland, and Erasmushogeschool Brussel, Vrije Universiteit Brussel, Belgium. She is a Researcher with CWC, University of Oulu. Her main research interests include lightweight security protocols, security and privacy on Internet of Things, and wireless sensor networks.

MIKA YLIANTTILA (SM'07) received the Ph.D. degree in communications engineering from the University of Oulu, in 2005. He was a Visiting Researcher with the Center for Wireless Information Network Studies, Worcester Polytechnic Institute, MA, and the Internet Real Time Laboratory, Columbia University, New York, USA. He is currently a Professor with the Centre for Wireless Communications, University of Oulu, and the Director of the Center for Internet Excellence, a research and innovation unit. He is also an Adjunct Professor of Computer Science and Engineering with the Faculty of Information Technology and Electrical Engineering. He has co-authored over 100 international peer-reviewed articles in broadband communications networks and systems, including aspects on wireless security, mobility management, distributed systems, and novel applications. He is an Editor of the *Wireless Networks* journal.

AN BRAEKEN received the M.Sc. degree in mathematics from the University of Gent, in 2002, and the Ph.D. degree in engineering sciences from the Computer Security and Industrial Cryptography Research Group, KU Leuven, in 2006. She worked for almost two years with BCG, a management consulting company. In 2007, she became a Professor with the Industrial Sciences Department, Erasmushogeschool Brussel, where she has been with Vrije Universiteit Brussel since 2013. Her current interests include cryptography, security protocols for sensor networks, secure and private localization techniques, and Field-programmable gate array implementations.

BURKHARD STILLER received the Computer Science Diploma and Ph.D. degrees from the University of Karlsruhe, Germany. He has been the Chair of the Communication Systems Group with the Department of Informatics, University of Zürich, since 2004. During his research with the Computer Laboratory, University of Cambridge, U.K., the Computer Engineering and Networks Laboratory, ETH Zurich, Switzerland, and the University of Federal Armed Forces, Munich, Germany, his main research interests cover charging and accounting of Internet services, economic management, systems with a fully decentralized control, telecommunication economics, and network and service management.

● ● ●