CHIP: Collaborative Host Identity Protocol with Efficient Key Establishment for Constrained Devices in Internet of Things

Pawani Porambage[1] · An Braeken[2] · Pardeep Kumar[3] · Andrei Gurtov[4] · Mika Ylianttila[1]

Abstract The Internet of Things (IoT) is the next evolutionary paradigm of networking technologies that interconnects almost all the smart objects and intelligent sensors related to human activities, machineries, and environment. IoT technologies and Internet Protocol connectivity enable wide ranges of network devices to communicate irrespective of their resource capabilities and local networks. In order to provide seamless connectivity and interoperability, it is notable to maintain secure end-to-end (E2E) communication links in

Part of this work has been published in IEEE Globecom, 2015 as 'Efficient Key Establishment for Constrained IoT Devices with Collaborative HIP-based Approach'. The proxy based key establishment component is taken from the paper which is entitled 'Proxy-based End-to-End Key Establishment Protocol for the Internet of Things' and presented at IEEE ICC Workshop on Security and Privacy for Internet of Things and Cyber-Physical Systems, 2015. The extensions of these works include the derivation of entire CHIP protocol, implementation it on Waspmote sensors, the measurement of energy costs, and the discussion of performance and security features in detail.

✉ Pawani Porambage
pawani.porambage@oulu.fi

An Braeken
an.braeken@vub.ac.be

Pardeep Kumar
pardeep.kumar@cs.ox.ac.uk

Andrei Gurtov
gurtov@acm.org

Mika Ylianttila
mika.ylianttila@oulu.fi

[1] Centre for Wireless Communications, Department of Communication Engineering, University of Oulu, P.o. Box 4500, 90014 Oulu, Finland

[2] INDI, Vrije Universiteit Brussel, 1000 Brussels, Belgium

[3] Department of Computer Science, University of Oxford, Oxford, UK

[4] Department of Computer and Information Science (IDA), Linköping University, Linköping, Sweden

IoT. However, device constraints and the dynamic link creations make it challenging to use pre-shared keys for every secure E2E communication scenario in IoT. Variants of Host Identity Protocol (HIP) are adopted for constructing dynamic and secure E2E connections among the heterogeneous network devices with imbalanced resource profiles and less or no previous knowledge about each other. We propose a solution called collaborative HIP (CHIP) with an efficient key establishment component for the high resource-constrained devices in IoT. CHIP delegates the expensive cryptographic operations to the resource rich devices in the local networks. Finally, by providing quantitative performance evaluation and descriptive security analysis, we demonstrate the applicability of the key establishment in CHIP for the constrained IoT devices rather than the existing HIP variants.

Keywords Internet of Things · Key establishment · Proxy · Host identity protocol · Resource-constrained devices

1 Introduction

Internet of Things (IoT) is the underlying fabric of next generation networking technologies that provide seamless connectivity among a broad range of smart objects under many application paradigms [1, 2]. A fundamental IoT technology, Machine-to-machine (M2M) communication extends the usability of wireless sensor networks (WSNs) to data communication between physical devices autonomously. In traditional WSNs, end-to-end (E2E) communication is considered only between the sensor nodes, which are deployed in a particular local network. However, the communication paths in M2M systems no longer follow the logical hierarchies and the topologies in the conventional WSN architectures [3]. Instead they advocate E2E communication among the sensor nodes and the remote hosts in distinctive networks [4]. On the other hand, unlike the centralized approach, in the distributed IoT architecture, the end devices are not dependent on a single central entity. The devices in distributed IoT are capable of acquiring data and services from other nodes in distinctive networks, and processing those retrieved data [5].

Securing IoT is also a key challenge due to the resource limitations, mobility and heterogeneity of the network devices [6]. The standardization organizations and professional associations such as Internet Engineering Task Force (IETF), European Telecommunications Standards Institute (ETSI), and Institute of Electrical and Electronics Engineers (IEEE) provide a noteworthy contribution to normalize IoT security standards. In certain IoT applications, the sensor nodes deployed in hazardous environments or battle fields that are difficult or impossible to access frequently, face the challenge of replacing batteries. Under such circumstances, it would be very critical to use high energy efficient security schemes in order to conserve the battery life.

The key establishment is a major prerequisite to construct a secure communication channel between two devices. Nevertheless, the resource consuming cryptographic operations on the caliber of the key establishment could be unaffordable or remarkably expensive to perform by a very wide range of resource-constrained devices in IoT networks [4]. This would bring an extra overhead to these devices and their normal operations since they exhibit constraints in both computational power and battery capacity. However, a key establishment occurs only at the initialization phase of a secure communication channel. Later on, the key can be reused until there is a necessity for rekeying. Therefore, a lengthy key establishment

process, such as few seconds, is still acceptable as long as it occurs once in a while during the entire operational mode [7].

Host Identity Protocol (HIP) is an IETF standard, that establishes secure signalling channels which inherently support node mobility and multihoming [8, 9]. With proper adaptations, HIP is also identified as a suitable key establishment protocol for securing IoT devices. Host Identity Indirection Infrastructure (HI3) is an example to a network architecture, which supports mobile hosts for secure mobility and multihoming [10]. HI3 is derived from Internet Indirect Infrastructure (I3) and HIP with better resilience and scalability. As a solution for the resource consuming cryptographic operations in the original HIP Base exchange (HIP BEX), the lighter version HIP Diet exchange (HIP DEX) was introduced to WSNs. However, HIP DEX still lacks the main security properties such as perfect forward secrecy and identity protection.

In order to provide E2E secure connectivity among the resource-constrained devices in IoT, we propose a solution called collaborative HIP (CHIP) with an efficient key establishment phase. According to CHIP, the highly constrained device delegates the computational resource demanding cryptographic operations of HIP protocol to resource rich devices in the neighborhood which are acting as proxies. Our main contribution is to define the design specifications and message exchange of CHIP, and provide a comprehensive performance and a security analysis. The significance is that we also address how CHIP adheres to the IoT network characteristics such as device heterogeneity, resource constraints, and unreliable communication links. Furthermore, we show quantitative performance evaluation for the proposed solution and compare the results with the energy costs and security properties for executing the key establishment of HIP BEX and HIP DEX variants.

The remainder of the paper is organized as follows: Sect. 2 provides background and related work about securing IoT and variants of HIP. Section 3 explains the network architecture, scenario description, and preliminaries that are used in describing the protocol. Section 4 gives a comprehensive description of the message exchange of CHIP protocol. Sections 5 and 6 present the performance and security analysis of the protocol. Section 7 provides a general comparison among CHIP, HIP DEX, and HIP BEX in consideration of their general attributes in the context of IoT applications. Finally, Sect. 8 summarizes the work and draws the conclusions.

2 Background and Related Work

In order to provide secure and seamless connectivity among IoT devices, it is noteworthy to ensure that the information exchanged in the network must be protected E2E. Therefore the basic security services such as confidentiality, authentication, and freshness of secret keys between two communicating entities should be carefully maintained [11]. Likewise, secure key management is identified as a major pilar of IoT security architecture [12]. The applicability of well-known E2E security mechanisms in the caliber of Transport Layer Security (TLS) or Internet Protocol Security (IPSec) is marginally suitable for the highly resource-constrained IoT devices such as sensors and actuators. Due to the lack of specifically tailored security solutions for constrained IoT devices, the standardization process is still not well established. Therefore, recent works by many standardization entities such as IETF, ETSI, and IEEE, focus on developing lightweight E2E security protocols for IoT [13, 14]. Over the past decade, several researches have been also performed to improve the E2E key establishment between resource-constrained IoT devices [4, 11, 15, 16].

Although the symmetric cryptography primitives based on pre-shared keys are low resource consuming, they are poorly applicable to the dynamic communication link creations between remote devices in IoT environments. Under such a circumstance, two party key agreement protocols are followed by the asymmetric key (or public-key) cryptographic primitives [4]. The most consistent candidates that have been currently proposed to establish E2E secured communications among the IoT devices, are Datagram Transport Layer Security (DTLS) handshake [17], minimal Internet Key Exchange (IKEv2) scheme [18], and Host Identity Protocol Diet Exchange (HIP-DEX) protocol [19]. HIP DEX and minimal IKEv2 mandate public-key cryptography in their protocol design whereas DTLS allows the options of using pre-shared-keys or public-key cryptography [20]. All the key agreements with the asymmetric cryptographic primitives have the variants of Diffie-Helman (DH) protocol [21]. The latest IETF standardization efforts are centered on DTLS since it has been chosen as the channel security underneath Constrained Application Protocol (CoAP) [13].

In [7], Said et. al. present an interesting idea about a collaborative key establishment protocol for resource-constrained IoT devices. Accordingly, a constrained device can delegate its heavy cryptographic load to less constrained nodes in the neighbourhood exploiting the spatial heterogeneity of IoT environment. Moreover, [7] shows how to integrate the collaborative approach with TLS handshake and IKE key establishment protocols. While initiating a secure E2E connection between two unknown nodes in distinctive networks, they exploit one set of intermediary nodes as proxies in order to support the key establishment process. However, this would not be realistic in the actual scenarios, since both the end nodes, which are completely unknown, may not have the securely pre-established communication links with those common proxies. As a solution for this, in [22] we have proposed a new key exchange scheme, which advocates a particular set of proxies for each end node.

Although HIP is identified as a key candidate for securing E2E connections in IoT, not many works are available for tailoring the HIP protocol in the context of IoT environments. HIP introduces a cryptographic namespace of stable host identities (HIs) between the network and transport layer [23]. Unlike DTLS and minimal IKEv2, HIP supports node mobility and multi-homing, which are very important attributes of IoT [24]. Following is a concise overview of HIP BEX and HIP DEX protocols and other lightweight HIP variants.

2.1 HIP BEX Protocol

As illustrated in Fig. 1 the message exchange of HIP BEX is a four-way handshake [9, 23]. The main objective of HIP BEX is to perform authenticated key agreement between two HIP peers (i.e., I for initiator and R for responder). The first I1 message sent by the initiator I, simply invokes the responder R to request a R1 message. The I1 message includes the source host identity tag (*SRC HIT*) and optional destination host identity tag (*DST HIT*). The responder replies with the R1 message, which is pre-computed and composed of a cryptographic puzzle, a public Diffie–Hellman key (DH_R), and a signature for node authentication. In order to continue with HIP BEX, the initiator has to solve the puzzle and provide the solution along with its public DH key (DH_I) and signature in I2 message. By solving and verifying the puzzle the initiator can convince its commitment to the responder to start secure communication and the responder can mitigate the denial-of-service attacks. In the mean time, the responder computes the DH session key K_{DH}. Once the responder has verified the solution, it can confidently continue the computation of the DH session key K_{DH} and start the secure HIP association with the initiator. The last message R2 in HIP

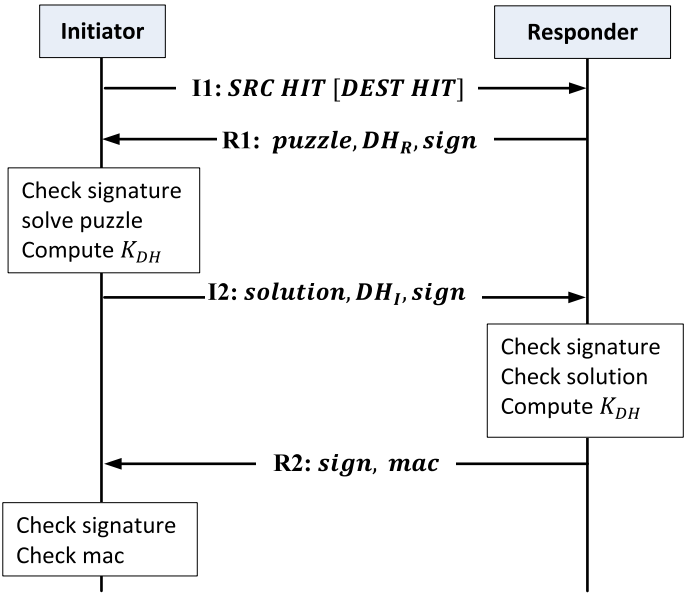

Fig. 1 Message flow of HIP BEX

BEX, finalizes the exchange and sends a (signed) message authentication code (MAC) computed with the generated DH key to the initiator for key confirmation. In HIP BEX, both, the initiator and responder undertake heavy cryptographic operations including the computation of two modular exponentiations for the generation of the DH key. Furthermore, the signature computations and verifications over R1, I2, and R2 messages are still not negligible resource consuming operations from the points of view of highly resource-constrained devices.

2.2 HIP DEX Protocol

HIP DEX is a modified version of HIP BEX protocol with reduced computational overhead. As explained in [8] and [19], the message flow of HIP Diet Exchange (HIP DEX) protocol is depicted in Fig. 2.

The significant differences with HIP BEX and HIP DEX, are the complete removal of the signature scheme in HIP BEX and the replacement of expensive DH key exchange with the Elliptic Curve Cryptographic (ECC) variant. The first message, I1, includes the source host identity tag (*SRC HIT*) and optional destination host identity tag (*DST HIT*). The second message R1 contains cryptographic challenge as a puzzle similar to HIP BEX, and public key PK_R. In HIP DEX, the initiator and responder perform Elliptic Curve DH (ECDH) key calculation by public key (PK) values to produce K_{DH}. The third message I2 contains the solution to the puzzle, PK_I and a key wrap parameter ($E(K_{DH}, x)$). The random values x and y are respectively the initiator's and responder's contributions to the final session secret key. The message I2 is also accompanied with the MAC value to ensure message integrity against tampering or corruption. The fourth message R2 also has the MAC value and the responder's key wrap parameters ($E(K_{DH}, y)$) and it finalizes the handshake. In HIP DEX, the ECDH key is used to encrypt the secrets (x, y), which will be

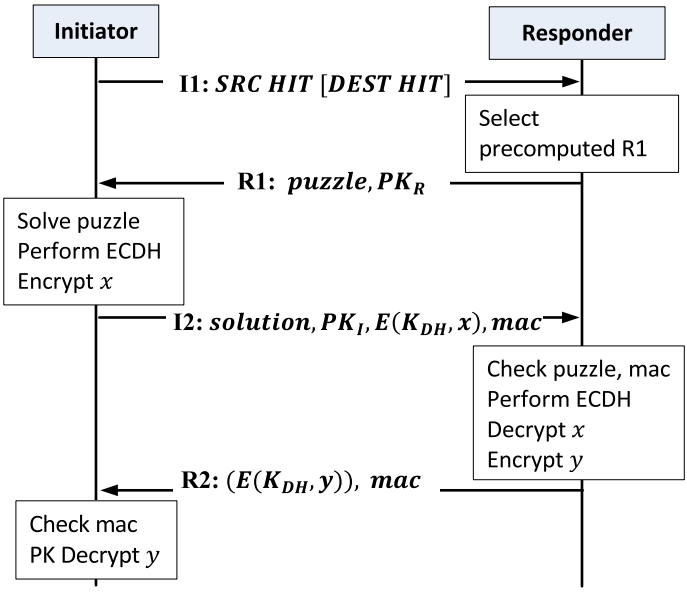

Fig. 2 Message flow of HIP DEX

eventually used to generate the final session key to encrypting subsequent data packets. In [14], the authors propose three extensions to lightweight HIP DEX including a comprehensive session resumption mechanism, a collaborative puzzle based DoS protection mechanism, and a refined retransmission mechanism.

2.3 Other HIP Variants

Moreover, lighter variants of HIP were discussed in several other publications [14, 25–27]. Lightweight authentication extension to HIP (LHIP) was proposed for CPU-restricted devices to access to HIP benefits such as end-host mobility and multihoming [27]. Although LHIP maintains a similar message syntax as in HIP BEX for compatibility reasons, it does not provide the same security level for host authentication and payload encryption. The main security drawbacks of LHIP are identified, since LHIP does not have DH key computation, RSA operation, and secure IPsec tunnel after the exchange. Instead, it uses hash chains for cryptographically binding successive messages with each other. However, this mechanism only guarantees the integrity protection over the current session.

As given in [25] and [26], distributed HIP (D-HIP) and Tiny HIP exchange have almost the same approaches to delegating resource-consuming cryptographic operations from the highly resource-constrained device (i.e., low performing initiator) to less constrained nodes in the neighborhood and establishing new HIP BEX connections with another unconstrained node (i.e., powerful responder). They exploit a common set of less constrained nodes as the proxies for supporting both the initiator and responder. Similar to the proxy-based solution in [7], this approach is not fitting either with the actual scenarios, due to the requirement of establishing secure links with a single set of proxies. This shortcoming is resolved in our previous work [28], which explains a collaborative HIP solution for high constrained devices with theoretical energy estimations. Similar to the solution given in

[22], we utilize two sets of proxies for two end nodes for delegating the cryptographic operations.

3 Network Architecture and Preliminaries

3.1 Network Architecture and Scenario Description

IoT facilitates heterogeneous networking technologies and devices with different capabilities in a wide range of applications including healthcare, surveillance, industrial and environmental monitoring [1]. In ambient assisted living (AAL) systems in IoT applications (e.g., E-health), there are several exemplary scenarios to describe the necessity of establishing secure E2E connections between two unknown resource-constrained devices [15].

As shown in Fig. 3, the scalar sensors are used to monitor the elderly person's health conditions. The sensors can be integrated into mobile personal devices such as smart phones, smart watches or even fixed to the human body as wearable devices. Whenever the scalar medical sensors receive health critical data, then they are responsible for invoking the visual sensors in their closest proximity. At the critical situations, the visual sensors can take the images of the patient and send to the responsible authorities (e.g., care taker, doctor, nurse). Since the person travels to different places (e.g., elderly home, hospital, library), the medical sensors may need to communicate with the visual sensors deployed in completely unknown environments. The visual sensors can be located in less accessible areas where frequent changing of the batteries is hard. Besides, handling multimedia traffic is also a very energy consuming task. Therefore, it is noteworthy to minimize the energy consumption for the cryptographic operations on the visual sensors. However, energy saving at the medical sensor side is less critical comparing to that since, the patient can recharge their batteries whenever needed. Consequently, establishing E2E secure

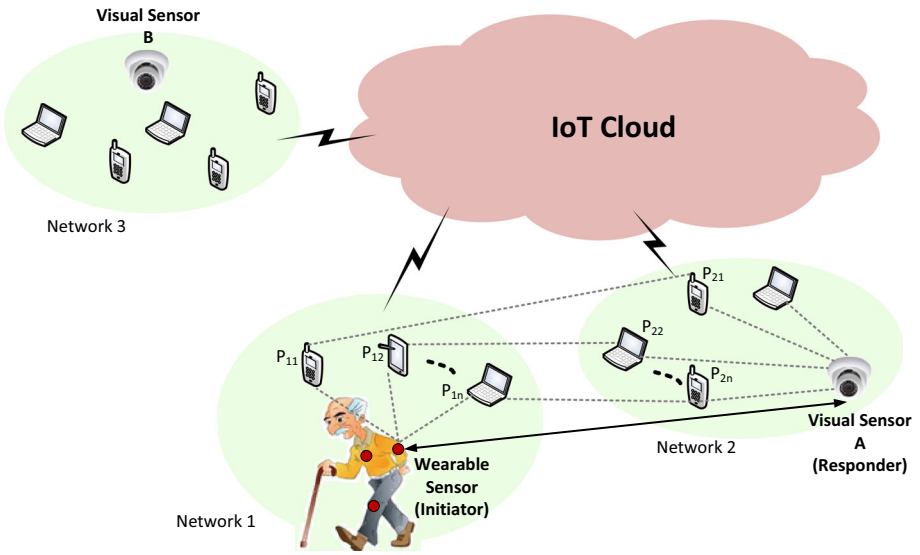

Fig. 3 Network architecture of an AAL system in IoT application

connection between the scalar and visual sensors in this particular scenario should always support node mobility and device power constraints.

As depicted in Fig. 3 we consider the E2E key establishment between two resource-constrained devices that have no previously shared keying materials. It is assumed that the highest energy efficiency should be obtained at the visual sensor. The initiator (i.e., wearable scalar sensor) has less resource constraints and energy saving requirements compared with the responder (i.e., visual sensor A). Each node is capable of formulating logical networks with a set of resource rich devices (e.g., smart phones, PDAs, laptops) in their neighborhood performing as proxies. This creates two logical networks (i.e., Network 1 and 2) and the proxies collaboratively support the two nodes for computing the shared secret key. It is assumed that the proxies in Network 1 and 2 can securely communicate with each other since they have enough resources and well-known techniques can be used. Furthermore, when the elderly person moves beyond the visual sensor A, the wearable sensor looses the contacts with A and it has to initiate a new connection with another visual sensor B, which is located in Network 3.

3.2 Preliminaries

3.2.1 Preparation of Involved Proxies

Proxies can be a set of neighboring nodes (e.g., smart phone, laptop, or PDA) which are resource-rich and have pre-established secure connections with the nodes (initiator and responder) [22]. The proper authentication and authorization proofs should be provided between the proxies and the nodes. Verification of the authorization of each proxy would be also resource consuming. Therefore, we consider that the nodes possess lists of neighbouring nodes in their closest proximity which have the potential of performing as proxies and the corresponding pre-shared keys for authentication. Since the number of those potential proxies is controlled, the storage consumed by those lists and keys will be also limited. However, the establishment of the secured connections between the two sets of proxies of the initiator and the responder sides, is out of the scope of this paper. This can be performed with the intervention of a trusted third party such as a trusted certificate authority. In order to simplify the notations, we provide a description where both communicate by means of symmetric key cryptography with secret shared keys established using well-known techniques (e.g., TLS or IPSec). For a particular CHIP handshake session, it is further assumed that the set of participating proxies for both end nodes (initiator and responder) will not change with time.

3.2.2 Definitions

The number of proxies contributing from each network is n. The protocol is based on a (n, k) threshold scheme [29], wherein n proxies process a polynomial share and k polynomial shares being enough to reconstruct the DH keys through the Lagrange polynomial interpolation. According to [29], the (k, n) threshold scheme is selected as $n = 2k - 1$. Note that k and n should be the same at initiator and responder, and $n \geq 3$. The initiator collaborates with the proxies $P_{11}, \ldots, P_{1n}$ whereas the responder collaborates with the proxies $P_{21}, \ldots, P_{2n}$. The initiator and the responder have the pre-shared keys with their corresponding proxies respectively known as $\{K_{i1}, \ldots, K_{in}\}$ and $\{K_{r1}, \ldots, K_{rn}\}$. Furthermore, the proxies from networks 1 and 2 have the shared secret keys such as $K_{p1}, \ldots, K_{pn}$

established through regular TLS or IPSec communication (e.g., shared key between the proxies P_{1i} and P_{2i} would be K_{pi}). As explained in [21] DH key exchange algorithm, we also consider the variable p is a prime and g is the generator in all the rest of the arithmetic operations.

4 Overview of CHIP Protocol

Message flow of CHIP protocol and the respective cryptographic operations are depicted in Fig. 4. Accordingly, the responder (i.e., the most resource-constrained device) delegates the most resource demanding cryptographic operations to the proxies while the initiator delegates only a part of those operations. Any proxy that supports either the initiator or the responder, does not have total knowledge about the subset P of participating-proxies. Therefore, although each proxy is contributing to computing one part of the DH key, none of them can collaborate with others to construct the DH key (K_{DH}) and thereby derive the final session key (i.e., derived by x and y).

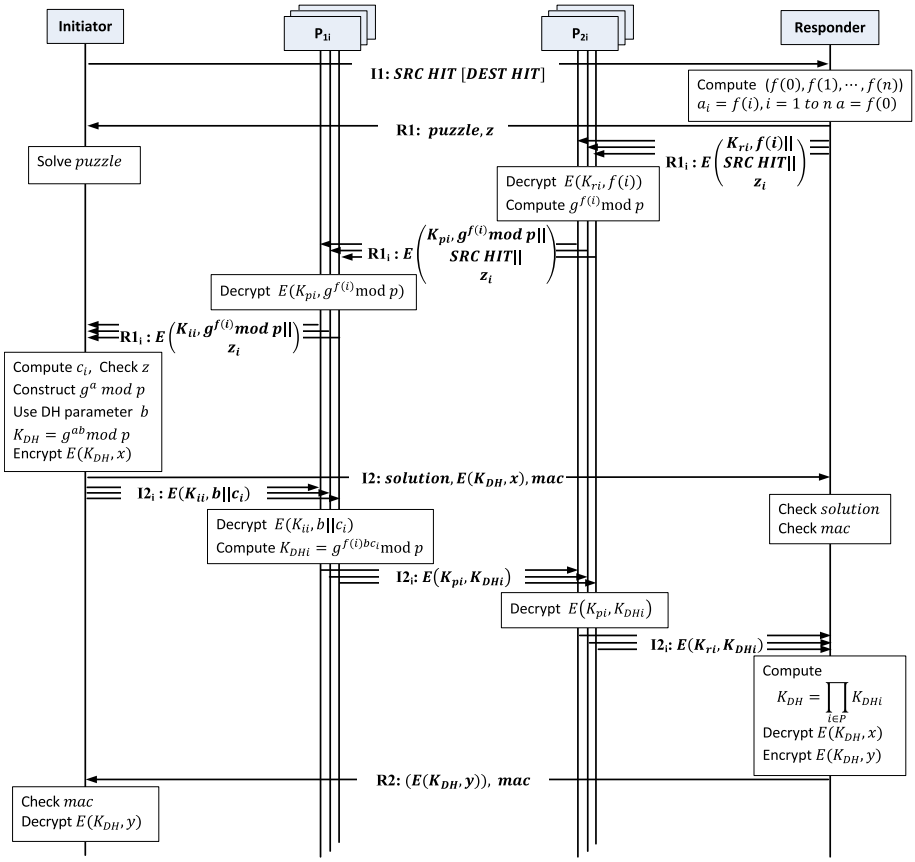

Fig. 4 Message flow of CHIP protocol

Step 1 First the initiator (I) starts communication by sending message *I*1 to the responder. The message I1 includes the *SRC HIT* and optional *DEST HIT*.

Step 2 The responder selects a pre-computed *R*1 message with a puzzle and secret z, and a set of $R1_i$ messages ($i = 1, \ldots, n$) where each contains a share a_i of the responder's DH private key a as follows. These distributed shares of the private exponent a are named as a_is and $a_i = f(i)$.

The derivations are referenced from [7] and [22]. Consider the polynomial function f of degree $k-1$ expressed as: $f(x) = q_0 + q_1 x + \ldots + q_{k-1}x^{k-1}$ where $q_1, q_2, \ldots, q_{k-1}$ are random, uniform, and independent coefficients. The responder's DH private key is a and $a = q_0$. According to Lagrange formula [29], the polynomial f can be derived as follows:

$$f(x) = \sum_{i=1}^{k} \quad f(i) \times \prod_{j=1, j\neq i}^{k} \frac{x-j}{i-j}\Bigg) \tag{1}$$

The responder calculates n values $f(1), f(2), \ldots, f(n)$ from the polynomial f where $n = 2k-1$ and $a = f(0)$. Each $R1_i$ message contains a share $f(i) = a_i$, *SRC HIT* and share z_i, which are encrypted by the pre-shared key K_{ri}. The z_i values are pre-computed as the reconstructing polynomial shares of the secret z. In accordance with Shamir's (n, k) threshold scheme, there are n number of z_i shares and only k values are required to obtain z. Then the responder sends $R1$ to the initiator and $R1_i$ messages to the corresponding proxies P_{2i}.

Step 3 Upon receiving $R1_i$ message, each proxy P_{2i} decrypts the message $D(K_{ri}, [E(K_{ri}, f(i)) \| SRC\,HIT \| z_i])$, computes its share of the responder's DH public key $g^{ai} \bmod p = g^{f(i)} \bmod p$, and forwards to a proxy P_{1i} in the initiator's network as an encrypted message $E(K_{pi}, g^{f(i)} \bmod p)$ along with the *SRC HIT* and z_i. By using the identity *SRC HIT*, the responder's proxy P_{2i} is capable of localizing the initiator's network and identify a potential proxy P_{1i}, who is willing to collaborate with the initiator and support the rest of the message exchanges.

Likewise, all the n number of proxies in the responder side collaborate with distinctive n proxies from the initiator side and maintain one-to-one mapping during the communication. This communication should satisfy an identity-based authentication, in order to ensure that not only one P_{2i} proxy of the responder can start a key request and not only one P_{1i} proxy of the initiator collects all the information and is able to take over the role of initiator.

Step 4 With the successful verification of $R1_i$ message, proxy P_{1i} decrypts the value $g^{f(i)} \bmod p$, encrypts the obtained value and z_i (i.e.,$E(K_{ii}, g^{f(i)} \bmod p \| z_i)$), and forwards the new $R1_i$ message to the initiator.

Step 5 Upon the reception of R1 message (i.e., from responder) and k number of $R1_i$ messages (i.e., from $P1_i$ proxies) from the subset P, the initiator first solves the puzzle and then starts computing the c_i coefficients as follows:

$$c_i = \prod_{i \in P, j \neq i} \frac{-j}{i-j} \tag{2}$$

Using the Lagrange formula and k number of c_i and z_i values, the initiator computes z^* and checks whether it is equal with the received z value.

$$z^* = \sum_{i \in P} c_i \times z_i \, mod \, p \tag{3}$$

This is an implicit assurance for the initiator to confirm that the intermediate proxies have accurately participated in the key establishment phase.

After that the initiator reconstructs the responder's DH public key using the Lagrange formula and the c_i values:

$$\begin{aligned} \prod_{i \in P} (g^{f(i)})^{c_i} \, mod \, p &= g^{\sum_{i \in P} f(i) \times c_i} \, mod \, p \\ &= g^{f(0)} \, mod \, p \\ &= g^a \, mod \, p \end{aligned} \tag{4}$$

Accordingly, the recipient (i.e., initiator) has to use only k successful deliveries out of n total messages for the consistent recovery of the responder's DH public key. Therefore, the protocol will remain uninterrupted in case of a proxy failure, misbehaving or unreliability and the loss of certain messages. Then the initiator derives the DH key $K_{DH} = (g^a \, mod \, p)^b \, mod \, p = g^{ab} \, mod \, p$ and uses it to encrypt the authenticated secret x, which will be used for constructing the final session key.

Later, the initiator computes message $I2$ and a set of $I2_i$ messages and respectively sends to the responder and the proxies P_{1i}s. The $I2$ message contains the solution to the puzzle, MAC value, and the secret x encrypted by K_{DH}, whereas each $I2_i$ message includes the encrypted b and c_i values $E(K_{ii}, b \| c_i)$.

Step 6 Similar to Steps 3 and 4, the intermediate proxies decrypt the $I2_i$ messages, compute the shares of the DH key (i.e., $K_{DHi} = g^{f(i)bc_i} \, mod \, p$) and forward them to the responder.

Step 7 After successfully receiving the $I2$ message and k number of $I2_i$ messages, the responder checks the solution and MAC values, and reconstructs the DH key K_{DH} as follows:

$$\begin{aligned} K_{DH} &= \prod_{i \in P} K_{DHi} \\ &= \prod_{i \in P} (g^{bc_i} \, mod \, p)^{f(i)} \\ &= g^{b \sum_{i \in P} f(i) \times c_i} \, mod \, p \\ &= g^{ab} \, mod \, p \end{aligned} \tag{5}$$

The responder uses K_{DH} to obtain x secret. Finally, the responder computes secret y, its authenticated share of the final session key, encrypts y by the DH key K_{DH}, attaches the MAC value, and sends to the initiator as $R2$ message.

Step 8 Once the initiator receives the R2 message, it verifies the MAC value, decrypts $E(K_{DH}, y)$, and obtains the secret share y.

After completing these eight steps, the initiator and the responder are able to construct the final session key using the secret shares x and y.

5 Performance Analysis

In order to evaluate the performance and quantify the energy efficiency of the key establishment component of the proposed CHIP protocol at the responder's side, we implement the corresponding cryptographic operations on Libelium Waspmote platform [30] using Waspmote cryptographic libraries. Waspmote has an Atmega1281 microcontroller running at 8 MHz with 8 KB SRAM, 4 KB EEPROM, and 128 KB Flash memory. We measure the execution time (t) for individual cryptographic operations and then calculate the computation energy cost using formula $U \times I \times t$ based on the execution time (t), the nominal voltage (U), and the current draw in active mode (I) on Waspmote sensors. As given in the data sheet [30], specifically we select $I = 9\,mA$ and $U = 3\,V$. We implement the CHIP protocol for $n = 5$ and $k = 3$ threshold scheme and use AES-128 for encryption and decryption (i.e., encrypted length is 16 *Byte*). ECC operations in HIP DEX are implemented for secp160r1 EC domain parameters (160-bit keys) and the protocol design specifications in [19]. In CHIP and HIP BEX approaches, size of the private exponent a and public key $g^a \, mod \, p$ are considered as 8 *Byte*. The resulted computation energy costs for the key establishment components of HIP variants with respect to the responder node are presented in Table 1.

The communication energy costs consist of the energy consumptions for transmission, reception, and listening. However, since the listening cost is very much dependent on the network size, processing time at the intermediate devices, and the nature of the communication links, we exclude it from the total communication energy cost. Therefore, we compute the total communication cost as a summation of transmission and reception energy consumption. According to [31], the communication delays to sending and receiving a packet of 100 Byte by Waspmote are respectively 11 and 50 ms. Consequently, transmission and reception energy consumption per byte are obtained as 2.97 and 13.5 µJ. Size of the sending and receiving messages by the responder in the key establishment phase of HIP variants are computed according to the exchange descriptions given in Sect. 4 and [9, 19]. Thereby we obtain Table 2 with total communication energy consumption for DH key derivation of HIP variants at the responder.

Bringing together the computation and communication costs, in Table 3, we provide the total energy costs for the key establishment phase of CHIP, HIP DEX, and HIP BEX protocols with respect to the responder node, which is considered the most constrained device. Accordingly, the key agreement phase in the CHIP protocol saves respectively 91% and 99% of energy at the responder node, compared with what is spent on the key agreement in HIP DEX and HIP BEX solutions.

Table 1 Computational energy costs for cryptographic operations of key establishment in HIP variants at responder

	Cryptographic operations	Energy cost
CHIP	Compute_$f(i)$_share $+ n*$encrypt_$f(i)$ $+ k*$decrypt_K_{DHi} $+$ Compute_K_{DH}	$13.5 + 18.9 + 3 \times 32.4$ $+ 5 \times 24.3 =$ **251.1 µJ**
HIP BEX [9]	Compute_K_{DH}	**237.816 mJ**
HIP DEX [19]	ECC_point_mult	**13.365 mJ**

Bold values are the final energy costs

Table 2 Communication energy costs for key establishment in HIP variants at responder side

	CHIP	HIP BEX [9]	HIP DEX [19]
Sent (byte)	80	8	20
Energy	237.6 μJ	23.76 μJ	59.4 μJ
Receive (byte)	48	8	20
Energy	648 μJ	108 μJ	270 μJ
Total communication energy	**885.6 μJ**	**131.76μJ**	**329.4 μJ**

Bold values are the final energy costs

Table 3 Total energy costs for key establishment in HIP variants at responder side

	CHIP	HIP BEX [9]	HIP DEX [19]
Computation energy	0.251	237.816	13.365
Communication energy	0.886	0.132	0.329
Total energy	**1.137 mJ**	**237.948 mJ**	**13.694 mJ**

Bold values are the final energy costs

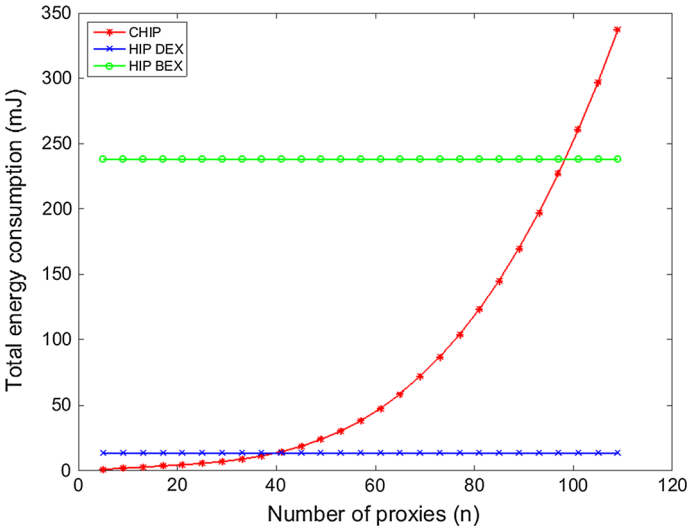

Fig. 5 Variation of total energy cost for key establishment with collaborating proxies (n)

As shown in Fig. 5, while increasing the number of proxies involved, the total energy cost for the key establishment component of CHIP grows exponentially.

The proposed key establishment phase outperforms than that of HIP DEX and HIP BEX when the cooperative proxies in one side are kept below 40. However, involvement of a very low number of proxies may also create a higher probability for them to communicate among the group of proxies and reconstruct the secret key. Therefore, it is important to maintain the balance between the number of involved proxies and the risk they tend to cooperate.

6 Security Analysis

The CHIP protocol proposed in this paper is adapted from the key design principles of HIP BEX and HIP DEX protocols. Therefore, similar to those original HIP protocols, our solution inherently withstand against replay and eavesdropping attacks [9]. In addition to that, there are few other security properties disclosed in the CHIP protocol as follows:

The perfect forward secrecy is a well-known security property of the DH key exchange protocol. Likewise, the utilization of ephemeral DH credentials in CHIP and HIP BEX protocols ensure this property. However, by using statically fixed keying materials for ECDH in HIP DEX, it losses the perfect forward secrecy property that is generally associated with DH protocol.

The proposed scheme takes the advantage of multiple deliveries of secret fragments in order to establish a HIP-based secure communication channel between two completely unknown devices in distinctive local networks. According to the Lagrange polynomial interpolation only k fragments out of n shares are required to compute the keys (i.e., responder's DH public key $g^a mod\, p$ at initiator and K_{DH} at responder). Therefore, from n total deliveries, only k successful deliveries are required for flawless calculations. This would eliminate the risk of the message losses in unreliable deliveries and the misbehaviour of some proxies such as refusing to cooperate, terminating operations, or compromising. Furthermore, each proxy receives one share of the secret (i.e., one out of n shares) and does not have the total knowledge about the entire set of proxies that are collaborating. Therefore, none of the proxies is able to derive the final key except the two end devices (responder and initiator). At least k proxies in each local network should be cooperative to derive the final key.

However, there is a risk that the proxies in each set may cooperate with each other by sharing different secret fragments delivered by the constrained nodes. This is not a threat to the overall security of the protocol, as long as the cooperating proxies are less than k. Assuming that proxies have equal likelihood to communicate among themselves, we calculate the probability P_{sec}, where less than k proxies can collaborate from n size set (where $n = 2k - 1$), as follows.

$$P_{sec} = \frac{\sum_{j=0}^{k-1} \binom{n}{j}}{\sum_{j=0}^{n} \binom{n}{j}} \tag{6}$$

The probability P_{sec} indicates the tendency where there is no adequate number of proxies cooperate to reconstruct the secret keys. This is also an indirect implication for the security of the protocol at each set of proxies (i.e., P_{2i}s and P_{1i}s). The variation of P_{sec} is shown in Fig. 6. Accordingly, while increasing n, P_{sec} goes to a maximum of 0.5, which is the assurance that the secret keys are completely secure.

In our solution, the initiator is capable of preventing Denial of Service (DoS) attacks, which can be created by the destructive proxies. Malicious proxy may try to disrupt the key establishment protocol by sending no or bogus traffic to the initiator. Therefore, in our solution, the initiator first calculates the z^* using the shares received by the proxies and then checks it with the received z value from the responder. If this validation is successful, the initiator can assure the reliability of the corresponding involved proxies and perform the rest of the cryptographic operations. Moreover, similar to HIP BEX and HIP DEX protocols, the CHIP protocol also provides a puzzle mechanism to mitigate packet DoS attack at the responder side.

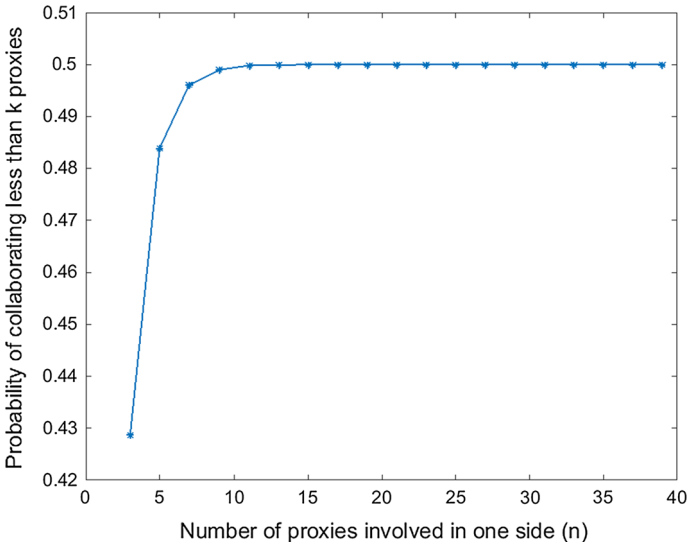

Fig. 6 Probability variation when the secret is secured

Unlike in HIP BEX, due to the absence of digital signature in HIP DEX protocol, the responder's identity cannot be verified by the initiator. Therefore, R1 message in HIP DEX is not protected and can be spoofed. Although, we completely eliminate the digital signature from R1 message in CHIP protocol, the responder's identity is still verified by the intermediate proxies. In order to proceed the communication (i.e., by sending $R1_i$ messages), the responder has to prove the proxies that itself is a legitimate node with a valid identity. Thereby, identity verification of the responder in CHIP is protected to a certain level.

During the E2E key establishment in CHIP, the protocol message transmission from one end to the other would occur by three intermediate steps (e.g., node-proxy, proxy-proxy, and proxy-node). As mentioned in the protocol, the constrained devices select a set of cooperative resource rich neighbours as proxies from their own local network. The communication links between the device and the proxies are already secured by the pre-shared keys and the devices can be authenticated within their local networks. This would create an implicit assurance that the nodes are legitimate inside the local network and the intermediate links facilitating the E2E connection establishment are secured. It also has the advantage of making man-in-the-middle (MITM) attacks more difficult between two constrained nodes. In every intermediate message, the useful information is encrypted with the corresponding keys in order to confirm the message confidentiality. Strict requirements for the secure communication between the proxies of initiator and responder (i.e., P_{1i}s and P_{2i}s) need to be made. During the communication, the identity of the proxies should be authenticated, in order to avoid MITM attacks of single proxy. Such an authentication process would be feasible at the proxies since they possess enough resources.

Table 4 summarizes the above-discussed security properties of CHIP, HIP BEX, and HIP DEX protocols.

Table 4 Security features of CHIP, HIP DEX, and HIP BEX

	CHIP	HIP BEX [9]	HIP DEX [19]
Perfect forward secrecy	Yes	Yes	No
Protection for replay attacks	Yes	Yes	Yes
Identity protection	Yes	Yes	No
DoS attack protection	Yes	Yes	Yes
MITM attack protection	Yes	Yes	Yes

Table 5 Comparison among CHIP, HIP DEX, and HIP BEX

	CHIP	HIP BEX [9]	HIP DEX [19]
Overhead	Low	High	Medium
Energy efficiency	High	Low	Medium
Security	High	High	Low
Mobility	High	High	High
Support device heterogeneity	Yes	No	No
Scalability in IoT	High	Low	Medium

7 Comparison

In order to understand the key differences and respective advantages, we provide a general comparison among CHIP, HIP DEX, and HIP BEX in Table 5.

In CHIP protocol, the computation overhead is minimized at the responder side by delegating resource consuming cryptographic operations to the less constrained devices. These operations include the computation of modular exponentiation (i.e., $g^a \, mod \, p$) and the DH key K_{DH} (i.e., $(g^a)^b \, mod \, p$) at the responder. Moreover, CHIP has a low overhead at the two end nodes since it does not use digital signatures and hash message authentication codes. HIP BEX has the highest overhead, due to the computation of two modular exponentiations and utilization of digital signatures, whereas HIP DEX has comparable less overhead due to the replacement of ECDH key establishment and removal of signatures.

According to the empirical results obtained in Sect. 5, CHIP shows the highest energy efficiency than the other two protocols. As explained in Sect. 6, CHIP and HIP BEX exhibit higher security properties than HIP DEX protocol. Mobility is an inherent feature of HIP, whereas all three variants discussed here possess this advantage by nature.

Unlike HIP BEX and HIP DEX protocols, CHIP takes the advantage of device heterogeneity of IoT nodes to establish secure E2E connections with the highly resource-constrained devices. In CHIP, the resource-rich devices involve as proxies and collaborate with the constrained peers. CHIP is designed in such a way to reused the existing secure connections to establish new communication links, and to operate broad ranges of devices. These features enhance the scalability attribute of CHIP in large IoT applications. Furthermore, similar to HIP DEX, in CHIP, there is no central element required, and no digital signature certificate needed. However, HIP DEX and HIP BEX show comparably low scalability in IoT applications due to their lower efficiency and lack of adaptability with the device heterogeneity. Thereby we can show that CHIP is out performing compared with HIP DEX and HIP BEX protocols in the context of highly resource-constrained devices in IoT.

8 Conclusions

This paper has proposed a proxy-based HIP protocol known as CHIP with an efficient key establishment scheme for creating secure E2E connections among the resource-constrained networking devices in the context of IoT. According to the syntax of the protocol although the two end nodes are completely unknown to each other, they have securely established connections with the proxies in their local networks. We have implemented different key establishment mechanisms of HIP variants on Waspmote sensors and obtained their energy consumption. The experimental results show that the proxy-based key establishment scheme in the CHIP protocol significantly increases the energy savings at the constrained responder compared with the standard HIP BEX and HIP DEX protocols. Consequently, in the performance and security analysis the proposed key establishment scheme generates significantly less computational overhead and energy consumption with stronger security features at the resource-constrained nodes than that of HIP BEX and HIP DEX protocols. Moreover, by introducing a digital signature scheme along with a similar proxy-based approach, we can further enhance the security strength of CHIP. Due to the extremely low energy profile and the consistency with the IoT network characteristics, a similar keying mechanism can be easily integrated with other IoT security schemes such as DTLS or IKEv2 protocols, for their better performance on resource-constrained devices. In future, we intend to extend the same proxy-based approach to securing multimedia traffic in IoT and optimizing energy utilization at the video terminals in wireless multimedia sensor networks.

Acknowledgements This works is supported by TEKES and the European Celtic-Plus project CONVINcE and was partially funded by Finland, France, Romania, Sweden and Turkey. A part of this research has been also supported by EU COST Action 1303 and the Naked Approach project funded by TEKES. Andrei Gurtov was supported by the Center for Industrial Information Technology (CENIIT).

References

1. Atzori, L., Iera, A., & Morabito, G. (2010). The Internet of Things: A survey. *Computer Networks*, *54*(15), 2787–2805.
2. Miorandi, D., Sicari, S., Pellegrini, F. D., & Chlamtac, I. (2012). Internet of Things: Vision, applications and research challenges. *Ad Hoc Networks*, *10*(7), 1497–1516.
3. Rawat, P., Singh, K., Chaouchi, H., & Bonnin, J. (2014). Wireless sensor networks: A survey on recent developments and potential synergies. *The Journal of Supercomputing*, *68*(1), 1–48.
4. Roman, R., Alcaraz, C., Lopez, J., & Sklavos, N. (2011). Key management systems for sensor networks in the context of the Internet of Things. *Computers and Electrical Engineering*, *37*(2), 147–159.
5. Roman, R., Zhou, J., & Lopez, J. (2013). On the features and challenges of security and privacy in distributed internet of things. *Computer Networks*, *57*(10), 2266–2279.
6. Roman, R., Najera, P., & Lopez, J. (2011). Securing the Internet of Things. *Computer*, *44*(9), 51–58.
7. Saied, Y. B., Olivereau, A., Zeghlache, D., & Laurent, M. (2014). Lightweight collaborative key establishment scheme for the Internet of Things. *Computer Networks*, *64*, 273–295.
8. Nie, P., Vähä-Herttua, J., Aura, T., & Gurtov, A. (2011). Performance analysis of HIP diet exchange for wsn security establishment. In *Proceedings of the 7th ACM symposium on QoS and security for wireless and mobile networks*, (pp. 51–56).
9. Jokela, R. M. P., & Melen, J. (2015). Using the Encapsulating Security Payload (ESP) transport format with the host identity protocol (HIP). *IETF RFC 7402*. http://tools.ietf.org/html/rfc77402.
10. Gurtov, A., Korzun, D., Lukyanenko, A., & Nikander, P. (2008). An efficient and secure networking architecture for mobile hosts. *Computer Communications*, *31*(10), 2457–2467.
11. Brachmann, M., Keoh, S. L., Morchon, O., & Kumar, S. (2012). End-to-end transport security in the ip-based Internet of Things. In *Porceedings of international conference on computer communications and networks*, (pp. 1–5).

12. Zhou, L., & Chao, H.-C. (2011). Multimedia traffic security architecture for the internet of things. *IEEE Network*, *25*(3), 35–40.
13. Keoh, S. L., Kumar, S., & Tschofenig, H. (2014). Securing the internet of things: A standardization perspective. *IEEE Internet of Things Journal*, *1*(3), 265–275.
14. Hummen, R., Wirtz, H., Ziegeldorf, J. H., Hiller, J., & Wehrle, K. (2013). Tailoring end-to-end ip security protocols to the Internet of Things. In *Proceedings of 21st IEEE international conference on network protocols (ICNP)*, (pp. 1–10).
15. Porambage, P., Schmitt, C., Kumar, P., Gurtov, A., & Ylianttila, M. (2014). PAuthKey: A pervasive authentication protocol and key establishment scheme for wireless sensor networks in distributed IoT applications. *International Journal on Distributed Sensor Netoworks*. doi:10.1155/2014/357430.
16. Porambage, P., Braeken, A., Kumar, P., Gurtov, A., & Ylianttila, M. (2014). Two-phase authentication protocol for wireless sensor networks in distributed IoT applications. In *Proceedings of IEEE wireless communications and networking conference (WCNC)*, (pp. 2728–2733).
17. Tschofenig, H., & Fossati, T. (2013). A TLS/DTLS 1.2 profile for the internet of things. IETF draft, RFC editor. http://tools.ietf.org/html/draft-ietf-dice-profile-09.
18. Kaufman, C. (2014). Internet key exchange (IKEv2) protocol. *IETF RFC 7296*. http://tools.ietf.org/html/rfc7296.
19. Moskowitz, R., Hummen, R. (2014) HIP Diet EXchange (DEX). *IETF draft*, RFC editor. http://tools.ietf.org/html/draft-moskowitz-hip-dex-02.
20. Kothmayr, T., Schmitt, C., Hu, W., Brünig, M., & Carle, G. (2013). DTLS based security and two-way authentication for the Internet of Things. *Ad Hoc Networks*, *11*(8), 2710–2723.
21. Diffie, W., & Hellman, M. (1976). New directions in cryptography. *IEEE Transaction on Information Theory*, *22*(6), 644–654.
22. Porambage, P., Braeken, A., Kumar, P., Gurtov, A., & Ylianttila, M. (2015). Proxy-based end-to-end key establishment protocol for the internet of things. In *Proceedings of IEEE ICC workshop on security and privacy for internet of things and cyber-physical systems*, (pp. 2677–2682).
23. Nikander, P., Gurtov, A., & Henderson, T. (2010). Host identity protocol (HIP): Connectivity, Mobility, multi-homing, security, and privacy over IPv4 and IPv6 networks. *IEEE Communications Surveys Tutorials*, *12*(2), 186–204.
24. Garcia-Morchon, O., Keoh, S. L., Kumar, S., Moreno-Sanchez, P., Vidal-Meca, F., & Ziegeldorf, J. H. (2013). Securing the IP-based Internet of Things with HIP and DTLS. In *Proceedings of the sixth acm conference on security and privacy in wireless and mobile networks*, (pp. 119–124).
25. Ben Saied, Y., & Olivereau, A. (2012) HIP Tiny Exchange (TEX): A distributed key exchange scheme for HIP-based Internet of Things. In *Proceedings of third international conference on communications and networking (ComNet)*, (pp. 1–8).
26. Saied, Y., & Olivereau, A. (2012). D-HIP: A distributed key exchange scheme for HIP-based Internet of Things. In *Proceeding of ieee world of wireless, mobile and multimedia networks (WoWMoM)*, (pp. 1–7).
27. Heer, T. (2007). LHIP lightweight authentication extension for HIP. *IETF draft*, RFC editor. http://tools.ietf.org/html/draft-heer-hip-lhip-00.
28. Porambage, P., Braeken, A., Kumar, P., Gurtov, A., & Ylianttila, M. (2015) Efficient key establishment for constrained IoT Devices with collaborative HIP-based approach. In *Proceedings of IEEE GLOBECOM*, (pp. 1–6).
29. Shamir, A. (1979). How to share a secret. *Communication ACM*, *22*(11), 612–613.
30. Waspmote sensor boards. Libelium. http://www.libelium.com/products/waspmote/.
31. Pham, C. (2014). Communication performances of IEEE 802.15.4 wireless sensor motes for data-intensive applications: A comparison of WaspMote, Arduino MEGA, TelosB, MicaZ and iMote2 for image surveillance. *Journal of Network and Computer Applications*, *46*, 48–59.