Securing IoT-based collaborative applications using a new compressed and distributed MIKEY mode

Mohammed Riyadh Abdmeziem

Securing IoT-based collaborative applications using a new compressed and distributed MIKEY mode

Mohammed Riyadh Abdmeziem

Inria-CNRS-LORIA,
Université de Lorraine,
Nancy, France
Email: mohammed-riyadh.abdmeziem@loria.fr

Abstract: Multimedia internet keying protocol (MIKEY) aims at establishing secure credentials between two communicating entities. However, existing MIKEY modes fail to meet the requirements of low-power and low-processing devices. To address this issue, we combine two previously proposed approaches to introduce a new compressed and distributed MIKEY mode applied to a collaborative internet of things context. A set of third parties is used to discharge the constrained nodes from heavy computational operations. Doing so, the MIKEY pre-shared mode is used in the constrained part of network, while the public key mode is used in the unconstrained part of the network. Furthermore, to mitigate the communication cost we introduce a new header compression scheme that reduces the size of MIKEY's header from 12 bytes to 3 bytes in the best compression case. To assess our approach, we performed a detailed security analysis using a formal validation tool (i.e., Avispa). In addition, we performed an energy evaluation of both communicational and computational costs. The obtained results show that our proposed mode is energy preserving whereas its security properties are preserved untouched.

Keywords: internet of things; IoT; collaborative applications; MIKEY protocol; key management protocols; security.

Reference to this paper should be made as follows: Abdmeziem, M.R. (xxxx) 'Securing IoT-based collaborative applications using a new compressed and distributed MIKEY mode', *Int. J. Information and Computer Security*, Vol. x, No. x, pp.xxx–xxx.

Biographical notes: Mohammed Riyadh Abdmeziem is a Lecturer_ Researcher (ATER) at Loria (University of Lorraine, Inria-CNRS-Loria, France), member of Coast team. Previously, he served as a postdoctoral fellow in the same team for two years. He received his PhD in Computer Science from the University of Sciences and Technology of Algiers (USTHB) in 2016 and was qualified for 'Maitre de conférences' from CNU (France) in 2019. His research activities are focused on security, confidentiality and key management protocols. He managed to publish several papers at an international level in conference proceedings, book chapters and journals.

This paper is a revised and expanded version of a paper entitled 'A new distributed MIKEY mode to secure e-health applications' presented at International Conference on Internet of Things and Big Data, Rome, Italy, 23–25 April 2016.

1 Introduction

Internet of things (IoT) is based on the pervasive presence of various wireless technologies such as radio-frequency identification (RFID) tags, sensors, actuators and mobile phones, in which computing and communication systems are seamlessly embedded (Lin et al., 2017). It is considered as one of the most important communication development in recent years. It makes our everyday objects (e.g., health sensors, industrial equipments, vehicles, clothes, etc.) connected to each other and to the internet (Abdmeziem et al., 2016a). Besides, collaborative peer-to-peer groups are increasingly popular, both in our personal and professional spheres. In fact, joining efforts to achieve common goals allows taking advantage of cumulated knowledge and experiences. Therefore, the cost to achieve these goals is considerably reduced, while the resulting quality is improved (Gnimpieba et al., 2015; Benouaret et al., 2013). In the context of ubiquitous IoT, collaborative applications are enhanced with contextual data gathered from the environment. This sensing is usually operated using tiny devices with highly constrained resources.

Establishing shared security credentials to secure communications between the constrained devices and the collaborative peers is challenging (Khan and Salah, 2017; Kouicem et al., 2018). In fact, it is daunting to consider existing key management schemes without introducing adapted mechanisms to take into account IoT specificities (Sicari et al., 2016). The scarcity of both power and computational resources will clearly hinder traditional solutions deployment (Jing et al., 2014; Tolone et al., 2005). In addition, considering the constrained nodes as part of the collaborative group induces significant security issues. Indeed, sensing devices are generally replaced after relatively short periods of time. Thus, authenticating each new device will generate a considerable overhead and will open security breaches. Moreover, the recurring rekeying operations to update the group key within the collaborative group will rapidly drain the device's battery energy level.

More importantly, engaging a highly constrained device with non-constrained entities in asymmetric cryptographic exchanges will inevitably lead to gaps in end to end security (Cheng et al., 2019). To address these issues, we extend two previous approaches (Abdmeziem et al., 2016b, 2018) to propose a new standard-based compressed and distributed key management scheme. Doing so, we design a new hybrid mode for MIKEY protocol that mitigates both computational and communication costs. Indeed, we do not consider the constrained node as part of the collaborative group, but rather, as a subcontractor acting on its behalf.

Firstly, we propose a new IPv6 over low power wireless personal area networks (6LoWPAN) header compression scheme to reduce the communication cost. Our scheme is intended to save energy and to avoid 6LoWPAN fragmentation that may occur when a datagram size exceeds the link layer MTU (Maximum Transmission Unit of the IEEE 802.15.4 protocol). Indeed, fragmentation is undesirable, as 6LoWPAN is vulnerable

to fragmentation attacks (Hummen et al., 2013b). Secondly, we propose a cooperative approach to discharge constrained nodes from heavy computational operations. To do so, we divide our network model into two segments. The first segment covers the communication channel between the constrained nodes and a set of third parties, to which the heavy computational operations are offloaded. To lighten the computational cost on constrained entities, only symmetric operations are used (i.e., pre-shared key mode). The second segment covers the communication channel between the third parties and any remote entity to which gathered data is transmitted. In this segment, asymmetric operations are used (i.e., public key mode).

The proposed distributed mode allows mitigating the disadvantages of both pre-shared key mode and the public key mode while benefiting from their advantages. To evaluate our proposed hybrid MIKEY mode, we first conducted a theoretical analysis of its security properties that were later formally validated through an implementation using Avispa tool (http://www.avispa-project.org). Second, we assessed the variation of the energy consumption of constrained entities using various numbers of third parties as well as numerous rates of compression. The obtained results showed that our approach does not alter the security soundness of MIKEY protocol, and all security features in terms of data confidentiality, data authentication, and data integrity are preserved. In addition, performances evaluation showed a remarkable gain in energy costs regarding both communication and computation overheads resulting from the introduction of distribution and compression.

The remaining of the paper is organised as follows. Section 2 provides the required background for a clear comprehension of the proposed approach. In Section 3, we introduce our compressed and distributed MIKEY mode. First, we present our network architecture and assumptions. Then, we detail the proposed approach. In Section 4, the security properties are analysed. Our performance analysis is presented in Section 5. Existing security solutions in the literature are surveyed in Section 6. Section 7 concludes the paper and sets our future research directions.

2 Background

In this section, we provide an overview of the features of MIKEY protocol (Arkko et al., 2004), while focusing on the adaptability of its different modes to constrained environments. In addition, we briefly present the concepts used throughout the remaining of the paper.

2.1 MIKEY overview

MIKEY is a key management protocol that aims to provide security associations to be used as an input for security protocols. The main motivation behind its design is to ensure end-to-end security while remaining simple and efficient (low-latency, low bandwidth consumption, low computational workload, small code size, and minimum number of roundtrips) (Arkko et al., 2004). The flexibility of MIKEY allows the designers to leverage upon several modes according to the specificities of the network scenario. Therefore, MIKEY seems to be the adequate protocol that can be extended to ensure secure communications in the collaborative IoT context. However, MIKEY various modes have not originally been designed to be implemented in constrained

environments with power and computation limitations, weak reliability of wireless links, and high scalability requirements.

Table 1 Terminology table

Notation	*Description*
I	Initiator
R	Responder
$data_k$	Data encrypted with key k
PSK	Pre-shared key
MAC	Message authentication code
PK_x	Public key of x
$CERT_x$	Certificate of x
TEK	Traffic encryption key
TGK	TEK generation key
$RAND$	Fresh value used for key generation
$auth\_key$	Authentication key
$encr\_key$	Encryption key
HDR	MIKEY header
T	Timestamp
ID_x	Identity of x
SP	Security policies
$KEMAC$	$\{TGK\}_{encr\_key/envelopekey}\|\|MAC$
PKE	$\{envelopekey\}_{PK\_R}$
$Sign_x$	Signature of x

MIKEY considers two entities that aim to establish a shared secret. One of the two entities assumes the *initiator* role, whereas the second one assumes the *responder* role. The key distribution modes are defined as follows (the different notations are described in Table 1):

- *Pre-shared key mode:* in this mode, both the *initiator* and the *responder* share a PSK from which two keys are derived, $encr\_key$ and $auth\_key$. An initialisation phase where the key is distributed is assumed. To establish a session, the *initiator* randomly generates a TGK, and sends it to the *responder* as part of the first message (i.e., I_MESSAGE). This latter is replay protected with timestamps, encrypted with $encr\_key$ and authenticated through a MAC using $auth\_key$. An optional verification response (i.e., R_MESSAGE) from the *responder* provides mutual authentication. R_MESSAGE contains a MAC computed upon both *initiator* and *responder* identities, and the same timestamp contained in I_MESSAGE using $auth\_key$ (Figure 1).

  In the pre-shared key mode, only symmetric operations are involved. This mode fits well, then, with the IoT constrained environment, as it can be run with limited energy and power resources. Nevertheless, this mode suffers from a severe scalability issue due to the pre-establishment phase during which a shared key is set between the involved parties.

- *Public key mode:* in this mode, the *initiator* transmits the generated TGK based on an 'envelope key' approach. The *initiator* encrypts and authenticates the TGK using a randomly/pseudo-randomly chosen envelope key, and sends it as part of I_MESSAGE. In addition, it includes the envelope key encrypted with the *responder* public key PK_R. In case where the *responder* owns several public keys, the *initiator* specifies the used key in the facultative CHASH parameter. Both ID_I and $CERT_I$ are also optional. It is worth mentioning that I_MESSAGE is signed using PK_I, and replay protected with timestamps. Similar to the Pre-shared key mode, an optional response message (R_MESSAGE) ensures mutual authentication (Figure 2).

  The public key mode is based on asymmetric primitives. These primitives use complex exponential operations, which prove to be difficult to run on constrained devices. On the other side, this mode does not require from the involved entities to pre-share credentials. Thus, two entities with no previous shared knowledge can establish a secure communication channel.

Figure 1 Pre-shared key mode signalling flow (see online version for colours)

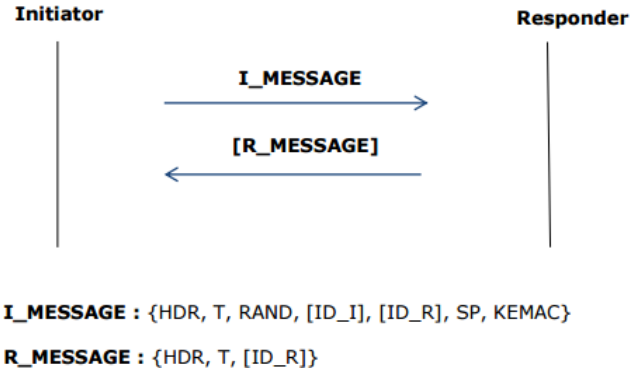
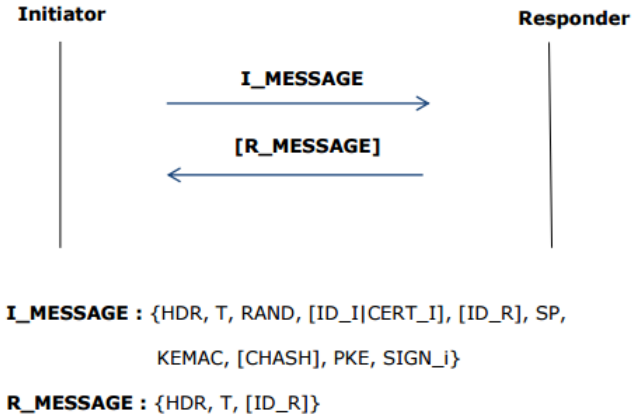


Figure 2 Public key mode signalling flow (see online version for colours)



In addition to the two previous modes, a third mode called 'Diffie-Hellman mode' is defined. This mode is mainly based on the Diffie-Hellman key exchange protocol. This mode has a higher computational and communication overhead compared to public key and pre-shared modes. Due to its inadequacy with our constrained scenario, this mode is ruled out.

2.2 Common header format (HDR)

The common header payload contains information about the different exchanged messages. It is included as the first payload within each message. In the following, we present a succinct description of each field contained in the MIKEY header. We refer to RFC 3830 (Arkko et al., 2004) for a more detailed description:

- *Version (8 bits):* version of MIKEY.
- *Data type (8 bits):* type of the exchanged message.
- *Next payload (8 bits):* identifies the payload added after the current payload.
- *V (1 bit):* flag to indicate the use of a verification message.
- *PRF func (7 bits):* indicates the key derivation function.
- *CSB ID (32 bits):* crypto session bundle (CSB) is a collection of one or more crypto sessions (CS). CSB ID field identifies the CSB.
- ♯ *CS (8 bits):* a crypto session refers to a data steam protected by a single instance of a security protocol. ♯ CS field indicates the number of crypto sessions within the CBS.
- *CS ID map type (8 bits):* specifies the method of uniquely mapping crypto sessions to the security protocol sessions.
- *CS ID map info (variable length)* identifies and maps crypto sessions to the security protocol sessions.

2.3 6LoWPAN adaptation layer

The 6LoWPAN standard defined in Hui and Thubert (2011) aims to transfer IPv6 packets through IEEE 802.15.4-based networks. 6LoWPAN uses IPV6 header compression mechanisms of IPv6 datagrams. Compression mechanisms are motivated by the limited space available in 802.15.4 frames to encapsulate IPv6 packets. In fact, the size of the 802.15.4 frame payload (102 bytes) leaves limited space for an IPv6 packet as 48 bytes are required merely for its header. 6LoWPAN defines encoding formats for compression based on shared state within contexts. In other words, it takes advantage of the fields that are implicitly known to all nodes in the network or can be deduced from the MAC layer. The compression scheme consists of IP header compression (IPHC) and next header compression (NHC).

IPHC encoding describes how an IPv6 header is compressed. 13 bits of the 2 bytes long IPHC are used for compression. The IPv6 header fields that are not compressed are placed immediately after IPHC. Moreover, NH field in IPHC indicates whether the

following header is encoded using NHC. If so, NHC encoding follows immediately the compressed IPv6 header. Compression formats for different next headers are identified by a variable ID bits plus the specific header compression encoding bits. The NHC to encode IPv6 extension headers and UDP header are already defined. For more details on 6LoWPAN, we refer to RFC 6282 (Hui and Thubert, 2011).

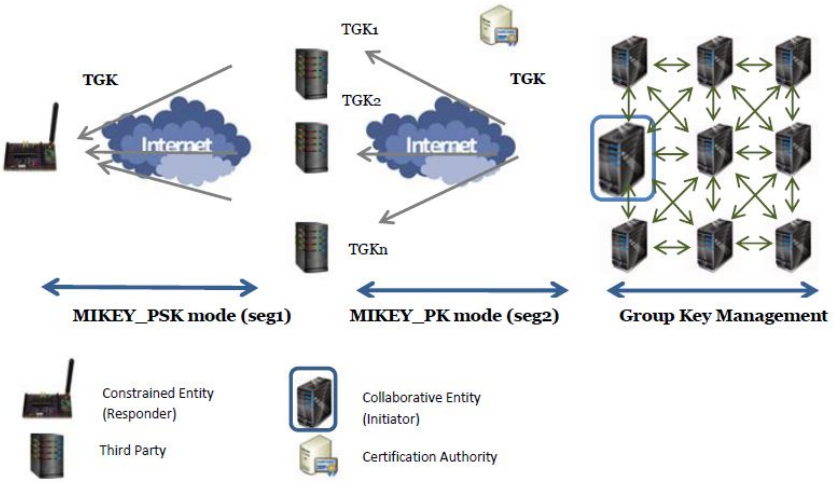
3 Contributions

In this section, we introduce a new compressed and distributed mode for MIKEY protocol. Firstly, we present our network architecture. Secondly, we define a set of assumptions before detailing our contributions.

3.1 Network architecture and assumptions

We consider an end-to-end secure communication channel between constrained smart objects (i.e., sensor nodes) and any remote unconstrained entity, which is part of a collaborative group. To do so, key management protocols are required between the two entities to secure their communications. These protocols have to deal with resources capabilities of the involved entities, along with the fact that no prior knowledge is established between them.

Figure 3 Compressed and distributed MIKEY mode: network architecture (see online version for colours)



IP-enabled smart objects are in charge of sensing data from the environment (e.g., temperature, pressure, stakeout, physiological data, industrial parameters, etc.). They are generally set in a remote location from the collaborative group. Gathered data is then transmitted from the smart object to a member of the collaborative group that is in charge of processing, analysing, and sharing data with other members of the collaborative group. In our architecture, we consider four main elements: the mobile and

contextual sensors, the third parties, the remote collaborative entity and the certification authority (Figure 3).

- *Mobile and contextual sensors:* the sensors are planted in the vicinity of the environment from which data is needed. This data is used during the collaborative process of the group.
- *Third party:* compared to the basic MIKEY modes, the third parties represent an additional component. A third party could be any entity that is able to perform high consuming computations.
- *Remote collaborative entity:* this entity receives the gathered data. Upon appropriate processing, the information is shared among the collaborative group.
- *Certification authority:* the certification authority is required to establish trust between the third parties and the remote entity by delivering valid and authenticated certificates.

The network is thus heterogeneous combining entities with various capabilities both in terms of computing power and energy resources. Smart objects (i.e., constrained sensors) have limited computational power, memory and energy resources. They are unable to perform public key cryptographic operations. However, the third parties and the remote collaborative entity are equipped with high energy, computing power and storage capabilities. They can take the form of server hardware or being distributed in a cloud infrastructure with flexible resources. The mapping with MIKEY concepts is defined as follows. *The initiator role* is mapped with the remote entity, which is part of a large collaborative group, while *the responder* is mapped with the smart object (also designated as constrained node).

Before presenting the details of our approach, we set the following assumptions:

- Constrained entities are able to perform symmetric encryption. Both third parties and the remote collaborative entity are able to perform asymmetric cryptographic operations.
- The third parties are not necessarily trusted.
- The certification authority is a trusted entity. It delivers authenticated cryptographic credentials.
- Each constrained node is able to keep a list of remote third parties. This list is pre-established during the initialisation phase.
- Each constrained node shares a PSK with each third party.
- Following its local requirements, the collaborative group selects a member, which will delegate data gathering tasks to an IP-enabled remote IoT object.

3.2 Reducing MIKEY communication overhead (compression)

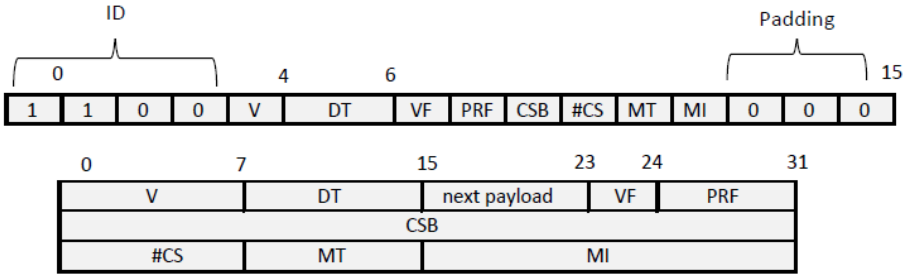
In this section, we describe our proposed 6LoWPAN header compression scheme for MIKEY. Our compression is based on the idea that the fields, which are implicitly known to all entities in the network or those that can be deduced from the MAC layer

can be omitted. As presented in Section 2.3, the NHC is used to encode the IPv6 extension headers and UDP header. Nevertheless, despite 6LoWPAN has defined header compression for UDP, no NHC compression is defined in case where headers contained in UDP payloads are compressed. In fact, MIKEY common header is contained in the UDP payload. Therefore, we propose to use the 6LoWPAN extension proposed in Raza et al. (2012a) to extend 6LoWPAN header compression mechanisms. These extensions indicate that the headers of protocols that are part of the UDP payload are compressed with 6LoWPAN-NHC.

Table 2 Gained space through the proposed MIKEY common header compression

Field (sizes in bits)	*MIKEY common header*	*Our 6LoWPAN-NHC-HDR*
Version (V)	8	1
Data type (DT)	8	2
Next payload	8	8
Verification V (VF)	1	1
PRF func (PRF)	7	1
CSB ID (CSB)	32	1
♯ CS	8	1
CS ID map type (MT)	8	1
CS ID map info (MI)	Variable length	1

Figure 4 Our 6LoWPAN-NHC-HDR encoding compared to the basic MIKEY header



MIKEY common header is 12 bytes long. It is appended to each packet through the different exchanged messages. We propose a 6LoWPAN-NHC to compress MIKEY header called 6LoWPAN-NHC-HDR. The proposed approach allows to reduce the header length from 12 bytes to 3 bytes (2 bytes for our 6LoWPAN-NHC-HDR plus 1 byte for the next payload field that is always carried inline) in the best compression case. In fact, only 13 bits are required to encode the different fields. Nevertheless, in order to remain standard compliant (i.e., the size of NHC encodings is multiple of bytes), our 6LoWPAN-NHC-HDR is 2 bytes long. In addition, to comply with 6LoWPAN-NHC encoding schemes, the first four bits implement an ID field to uniquely identify our NHC encoding. We set the ID bits to 1100. To the best of our knowledge, the 1100 bits are currently unused as NHC identifiers. In the following, we present in detail the encoding approach for each field (see Table 2 and Figure 4):

- *Version (V):* if 0, the version is the default and latest MIKEY version defined in Arkko et al. (2004) and the field is skipped. If future versions are defined, the bit is set to 1 and the version number is carried inline after the 6LoWPAN-NHC-HDR header. Our compression is thus kept dynamic and flexible.
- *Data type (DT):* the data type field describes the type of the exchanged messages. Based on our proposed distributed mode (see Section 3.3), we only consider three types of messages (plus the ERROR type), which are involved with the constrained nodes. Doing so, we are then able to use just 2 bits encoding for the data type field instead of 8 bits in the original MIKEY modes:

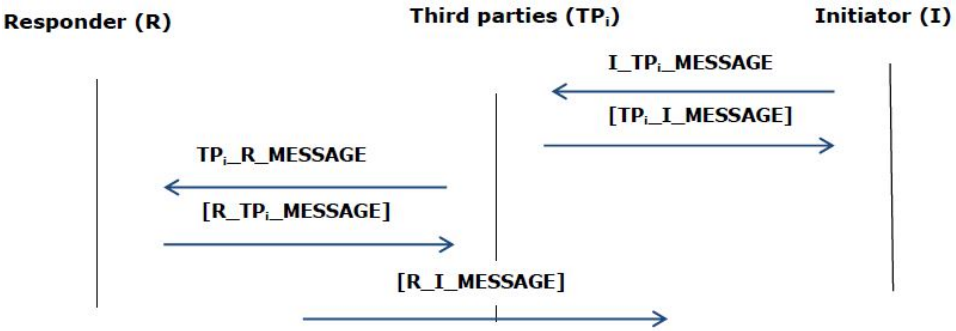
  00 : I_TPi_MESSAGE
  01 : TPi_I_MESSAGE
  10 : R_I_MESSAGE
  11 : ERROR
- *Verification V (VF):* the VF field encoding is similar to the non-compressed header. If it is set to 0, no verification message is used. When it is set to 1, a verification message is required.
- *PRF func (PRF):* if 0, the default PRF function defined in Arkko et al. (2004) is used. If set to 1, the PRF function value is carried inline.
- *CSB ID (CSB):* the CSB ID is chosen by the *initiator* and needs to be unique between each *initiator-responder* pair. Instead of carrying its 32 bits size inline, we propose to derivate the CSB ID from the concatenation of lower layer identifiers (e.g., IPv6 addresses). One bit is sufficient for the encoding. If set to 0, the CSB ID is derived instead of being carried inline. If set to 1, the 32 bits CSB ID are carried after the 6LoWPAN-NHC-HDR header.
- ♯ *CS:* if we assume in our constrained scenario that there is only one CS in each CSB, there is no need therefore for keeping 8 bits to indicate the number of crypto sessions. We are then able to encode the ♯ CS with 1 bit. If this bit is set to 0, only one CS is considered. In addition, to make our compression flexible, if the bit is set to 1, the number of CS is carried inline.
- *CS ID map type(MT):* if 0, the default GENERIC-ID map type defined in Arkko et al. (2004) is used. If set to 1, the CS ID map type is carried inline.
- *CS ID map info (MI):* the CS ID map info size is kept variable in Arkko et al. (2004). If we assume that there is only one CS in each CSB, we could use 1 bit for the encoding. If 0, the unique CS is identified with its corresponding mapping to the security protocol for which security associations are created. If set to 1, the map info field is carried inline.

The next payload field is always carried inline as it is impossible to predict or deduce the next payload content. In addition, the three last bits are used as padding bits to remain standard compliant with RFC 6282 (Montenegro et al., 2007) (NHC size is defined as 2 bytes long).

3.3 Reducing MIKEY computation overhead (distribution)

We divide our network architecture into two segments. The first segment is defined by the communication channel linking the constrained node to the third parties. This segment involves the constrained entities of our network model. As a result, we consider using the pre-shared key mode. The second segment is defined by the communication channel linking the third parties to the collaborative entity. This segment does not suffer from resources constraints. As a result, we consider using the public key mode.

Figure 5 Illustration of the different message exchanges of our proposed mode (see online version for colours)



After an initialisation phase where the constrained node is pre-loaded with a set of third parties identities, along with the different PSK, our distributed mode proceeds with successive messages. Table 1 summarises the notations used, and Figure 5 illustrates the signalling flow. To remain standard compliant, the messages header, along with various message parameters are kept unchanged [RFC 3830 (Arkko et al., 2004)]. In the following, we detail the different exchanged messages:

- *I_TPi_MESSAGE:* the *initiator* randomly generates a secret TGK, which will be used later to further derive keying materials at both I and R sides. The TGK is split into *n* parts TGK_1, TGK_2, ...TGK_n. I randomly generates an envelope key. This key is used to encrypt and authenticate the generated TGK_i parts, which are included in *I_TPi_MESSAGE*. The envelope key is encrypted with the public key of each TP_i and included in the message. In addition, I's signature that covers all the fields of the message is also included. Each part is then sent to the appropriate TP_i in *I_TPi_MESSAGE*. The general structure of the message is as follows:

  $\forall i \in \{1, N\}\ \{HDR, T, RAND, [ID_I], [CERT_I],$
  $[ID_R], SP, KEMAC_i, [CHASH]\}_{PK_{TP_i}}, PKE, SIGN_I$

  Because wireless connection is the main media in IoT, I applies an error redundancy scheme to the generated TGK. Doing so, R can retrieve the secret without requiring the reception of all the packets, if some of them were lost during the transmission process. For instance, the widely used Reed-Solomon scheme can be applied (Reed and Solomon, 1960).

- *TPi_I_MESSAGE:* upon receiving *I_TPi_MESSAGE*, each TP_i authenticates and decrypts the received message. An optional verification response sent from TP_i to I provides mutual authentication. The structure of the message is as follows.

  $\forall i \in \{1, N\}\ \{HDR, T, [ID_R]\}_{PK_I}$

- *TPi_R_MESSAGE:* after having properly authenticated and decrypted the received *I_TPi_MESSAGE*, TP_i includes TGK_i in *TPi_R_MESSAGE*. This message is replay protected with timestamps, encrypted and authenticated using the pre_shared PSK. The structure of the message is as follows:

  $\forall i \in \{1, N\}\ \{HDR, T, RAND, [ID_I], [ID_R], SP\}_{PSK_i}, KEMAC_i$

- *R_TPi_MESSAGE:* upon successful authentication and decryption of *TPi_R_MESSAGE* by R, the TGK is retrieved. In fact, after having received enough packets containing the different TGK_i, R reconstructs the original TGK. An optional verification response sent from R to TP_i provides mutual authentication. The structure of the message is as follows:

  $\forall i \in \{1, N\}\ \{HDR, T, [ID_R]\}_{PSK_i}$

- *R_I_MESSAGE:* using the established TGK, R encrypts and authenticates a verification message (i.e., *R_I_MESSAGE*). This latter is sent to I, which authenticates the received message. A successful authentication is considered as a proof of R's knowledge of TGK. It is worth noting that *R_I_MESSAGE* is optional and only sent if ID_I has been included in the different exchanges. The structure of the message is as follows:

  $\{HDR, T, [ID_R]\}_{TEK}$

The reconstructed TGK is used to derive further keying materials. The derivation process is detailed in MIKEY RFC 3830 (Arkko et al., 2004). Both I and R are then able to derive state connection keys for encryption and authentication of the exchanged data. A secure end-to-end channel is hence created between highly constrained devices (i.e., sensors) and remote unconstrained entities. Our distributed MIKEY mode takes advantage of both pre-shared and public-key modes, while mitigating their disadvantages.

4 Security analysis

4.1 Key exchange properties

In this section, we briefly analyse the security features of our proposed mode based on the properties presented in Roman et al. (2011). For the following discussion, we consider our communication channel split into two segments: Seg1 from R to the TP_i and Seg2 from the TP_i to I (see Figures 3 and 5):

- *Confidentiality:* regarding Seg1, the exchanged messages between R and the different TP_i are encrypted using the corresponding PSK_i. Based on RFC 3830 (Arkko et al., 2004), we advocate the use of AES-CCM mode that defines

AES-CBC for MAC generation and AES-CTR for encryption (Dworkin, 2007). Nowadays, more and more tiny sensors include AES hardware co-processor, which would help to decrease the overhead. Regarding Seg2, communications are secured using Public Key Encryption. The certification authority is in charge of delivering the required certificates.

- *Authentication and integrity:* in our protocol, communications are authenticated using MACs in Seg1 and digital signatures in Seg2. The exchanged data is, then, guaranteed to remain genuine. This property ensures that data has not been altered, and has been sent from legitimate entities (and to legitimate entities, as verification messages can be added to provide mutual authentication). Furthermore, nonces (i.e., time-stamps) are included in the exchanged messages for protection against replay attacks.
- *Distribution:* similar to the Pre-shared mode, an initialisation phase is required to distribute the shared PSK between the constrained nodes and the TP_i. This phase is generally performed off-line. Nevertheless, in Seg2 and similar to the Public key mode, TP_i and I establish a secure channel in an online mode taking advantage from the asymmetric primitives. As a consequence, upon an initial distribution in Seg1, our proposed mode can be run without any external intervention allowing automatic updates.
- *Overhead:* the constrained entities are only involved in symmetric operations, which are much less resource consuming than asymmetric ones (Wander et al., 2005). Actually, the powerful third parties take in charge all asymmetric operations. Indeed, limiting computation solicitations for the constrained nodes decreases their power consumption and thus increases their battery life-time.
- *Resilience:* involving several third parties in the key exchange process makes our protocol highly resilient. To compromise and recover the exchanged secret TGK, an attacker would need to corrupt all third parties, as TGK is split into numerous shares. Thus, unless an attacker compromises all TP_i, it is nearly impossible to recover the original TGK. As a result, our mode does not assume all third parties to be trusted.
- *Extensibility and scalability:* in a collaborative IoT scenario, contextual sensors can be needed at any time to gather specific data. Our protocol requires an initialisation phase where the sensor (i.e., R) is set with a list of TP_i identities along with the PSK_i that are shared with each TP_i. However, our protocol proceeds without any operation regarding the TP_i or I. After the initialisation phase, the joining node is ready to establish an end-to-end secure channel with any remote entity.
- *Storage:* due to recent hardware advances in flash memory (Tsiftes and Dunkels, 2011), smart objects provide considerable amounts of storage space. This space is used to store the TP_i's identities list, along with the corresponding PSK_i. Furthermore, we assume that the number of TP_i will not exceed a reasonable threshold. Consequently, storage space is not considered as an issue in our protocol deployment.

4.2 Formal validation

To prove that our protocol does not violate the required security properties, in particular, confidentiality, authentication, delivery proof and replay protection, we carried out an analysis using Avispa tool (http://www.avispa-project.org). Automated Validation of Internet Security Protocol and Applications (AVISPA) is a state-of-the-art verification tool for security protocols that includes a set of model checkers with a common front end. The tool follows the Dolev-Yao intruder model (Dolev and Yao, 1981) to intercept messages, or to insert modified data. It performs analytical rules to state whether the protocol is safe or not. In case of unsafety, the tool provides a trace highlighting the steps that led to the attack.

Figure 6 Avispa output ($OFMC$) (see online version for colours)

```
user@instant-contiki:~/HybridMIKEY$ avispa HybridMIKEY.hlpsl --ofmc
% OFMC
% Version of 2006/02/13
SUMMARY
  SAFE
DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
PROTOCOL
  ../avispa-1.1/testsuite/results/HybridMIKEY.if
GOAL
  as_specified
BACKEND
  OFMC
COMMENTS
STATISTICS
  parseTime: 0.00s
  searchTime: 103.66s
  visitedNodes: 13400 nodes
  depth: 18 plies
```

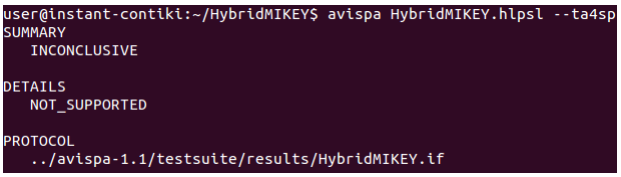
Figure 7 Avispa output ($CL - AtSe$) (see online version for colours)

```
user@instant-contiki:~/HybridMIKEY$ avispa HybridMIKEY.hlpsl --cl-atse

SUMMARY
  SAFE

DETAILS
  BOUNDED_NUMBER_OF_SESSIONS
  TYPED_MODEL

PROTOCOL
  ../avispa-1.1/testsuite/results/HybridMIKEY.if

GOAL
  As Specified

BACKEND
  CL-AtSe

STATISTICS

  Analysed   : 2290 states
  Reachable  : 1114 states
  Translation: 0.15 seconds
  Computation: 0.59 seconds
```

Protocol models in Avispa are written in a role-based language called High Level Protocol Specification Language, or HLPSL (Chevalier et al., 2004). The actions of the different entities are specified in a module called *basic role*, while their interactions are defined by composing multiple *basic roles* together into a *composed role*. In addition, the security goals of the analysed protocol are specified in the *goal section* before launching the analysis. Besides, Avispa uses several different automatic protocol analysis

techniques to validate the analysed protocol against the specified security goals, such as the on-the-fly model-checker (OFMC), and the constraint-logic-based attack searcher (CL-AtSe).

Figure 8 Avispa output ($TA4SP$) (see online version for colours)



In our modeling, we first specified a *basic role* to describe the actions of the different entities involved. Then, we specified how the participants interact with each other in a *composed role*. The different roles were specified using the HLPSL language, and introduced as an input for Avispa tool. The specification has been analysed against the Dolev-Yao intruder model using the OFMC, the TA4SP, and the CL-AtSe backends. The results were indicated in reports for each backend model produced by Avispa tool. They show that our new exchange mode is 'SAFE' against OFMC (Figure 6), and $CL-AtSe$ (Figure 7). However, against TA4SP database, the result was 'INCONCLUSIVE' (Figure 8). According to Avispa user manual (http://www.avispa-project.org), an inconclusive result does not imply that an attack has been detected. Based on the obtained results, we can affirm that our proposed hybrid mode is safe with respect to the specified security goals.

5 Performances analysis

In this section, we provide a performance evaluation of our contribution focusing on energy consumption. Indeed, battery-powered IoT devices are highly sensitive to energy limitations. We proceed by presenting the energy model upon which our evaluation is based, then we discuss and analyse the obtained results.

5.1 Energy model

To assess the energy consumption of our tailoring approach of MIKEY protocol, we defined an energy model. This model is based on the measurements presented in De Meulenaer et al. (2008) and Kaps and Sunar (2006). In De Meulenaer et al. (2008), authors measured the energy consumption triggered by communication operations, whereas in Kaps and Sunar (2006), authors measured the energy consumption triggered by computation operations. These measurements took place on constrained entities providing only few MHz of computational power, and few kilobytes of RAM and ROM. In our evaluation, we consider the total energy cost as the sum of both computational and communicational costs. In fact, the communicational cost is the result of sending and receiving operations, while the computational cost the result of authentication and encryption operations. The energy model is presented in Table 3.

Table 3 Energy model

Operation	*Cost*
Send 1 bit	0.72 μJ
Receive 1 bit	0.81 μJ
AES-128 128-bits encryption	28.11 μJ
SHA-1 128-bits MAC computation	23.9 μJ
Public key encryption (ECC-160)	17 mJ
Public key signature (ECDSA-160)	15 mJ

Besides, we set the following assumptions regarding our evaluation:

- We only focus on the energy constrained parts of our network architecture, namely, the mobile and contextual sensors.
- We only consider the header in the exchanged messages. The header is the part that is subject to our compression scheme. The remaining components of the messages are constant in term of size for both basic and tailored MIKEY protocol.
- The length of the 'CS ID map info' field is dynamic and is not set in MIKEY specification. Indeed, to perform our estimation, we set the size to 2 bytes.

5.2 Results

To assess the energy gains obtained through our approach, we first evaluated the energy cost of a MIKEY key exchange session when no third party is used. In other words, using asymmetric primitives. Then, we evaluated the energy cost while varying the number of third parties from two third parties to ten. In addition, we considered numerous header compression rates when evaluating the energy consumption ranging from 0% (no compression) to 100 % (maximum compression). Table 4 indicates the different used compression rates. Each rate determines the set of fields to be compressed using our proposed 6LoWPAN-NHC-HDR.

Table 4 Space gained vs compression rate

Compression rate (%)	*Compressed fields*	*Gained space (bits)*
0	None of the fields are compressed	0
16.4	V, DT	13
32.9	V, DT, PRF, MT	26
51.9	V, DT, PRF,♯ CS, MI	41
72.1	V, DT, PRF, MT, CSB	57
83.5	V, DT,♯ CS, MI, CSB	66
100	All the fields are compressed	72

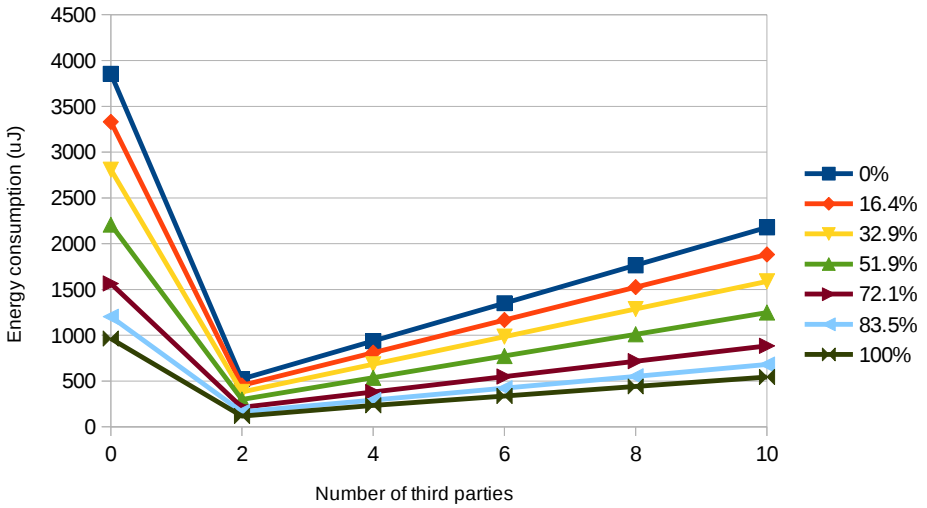
The results of our performances analysis are introduced in Table 5. In Figure 9[1], we depict the evolution of the energy consumption while increasing both the compression rate and the number of third parties. It is clear that energy consumption drops sharply with the introduction of two third parties. This is due to the fact that using third

parties spares the constrained nodes from running asymmetric primitives, which are much resource consuming than symmetric primitives. Furthermore, we note that the energy consumption increases with the inclusion of additional third parties, albeit the energy consumption remains far inferior to the case where no third party is used. The increase in energy consumption with the introduction of third parties is the result of an increased number of exchanged messages, which raises the overall energy consumption overhead. Furthermore, the results show a decrease in energy consumption each time the compression level is raised. This is due to the fact that both communicational and computational costs are reduced when the size of data to send, receive, encrypt, or authenticate is reduced (i.e., compressed).

Table 5 Energy cost evaluation

Compression (%)	*# Third parties*	*Communication (uJ)*	*Computation (uJ)*	*Total cost (uJ)*
0	0	146.88	38,400	38,546.88
	2	362.88	159.16	522.04
	6	950.4	399.48	1,349.88
	10	1,537.92	639.8	2,177.72
51.9	0	84.15	21,990	22,074.15
	2	207.9	91.13	299.03
	6	544.5	228.7	773.2
	10	881.1	366.33	1,247.43
100	0	36.72	9,600	9,636.72
	2	90.72	26.77	117.49
	6	237.6	99.87	337.47
	10	384.48	159.95	544.43

Figure 9 Compressed and distributed MIKEY mode: energy consumption evolution (see online version for colours)



Relying on the performance evaluation results, we can conclude that our proposed hybrid MIKEY mode provides a noticeable gain in energy consumption that is highly sought by battery-powered constrained IoT-based entities. As a matter of fact, both compression and distribution contribute in reducing MIKEY overhead. The more compression rate is increased, the more energy gain is obtained. It is worthy of note that the number of header fields which can be compressed depends on the application for which MIKEY protocol is implemented. Besides, including additional third parties strengthens security, as more of them need to be corrupted in order to retrieve the exchanged secret materials. However, this result in less energy gains due to the communication overhead. Hence, a trade-off is required between security and performances.

6 Literature review: discussion

In our literature review, we distinguish two main research axes. The first axis is focused on compression schemes applied on standard-based protocols, while the second axis is focused on approaches based on the offloading of heavy computational operations to third parties. Numerous energy aware approaches have been introduced for the IP-based IoT (e.g., Abdelfadeel et al., 2017; Raza et al., 2012b; Hummen et al., 2013a). In Montenegro et al. (2007) and Hui and Thubert (2011), the compression of IPV6 headers, extension headers along with UDP headers has been standardised through 6LoWPAN. Raza et al. (2011) presented 6LoWPAN compressions for IPsec payload headers (AH and ESP). In Raza et al. (2012b), an IKE compression scheme has also been proposed providing a lightweight automatic way to establish security associations for IPsec. Likewise, header compression layers for DTLS, HIP DEX, and HIP BEX were respectively introduced in Raza et al. (2012a), Hummen et al. (2013a) and Sahraoui and Bilami (2015). Furthermore, Abdmeziem et al. (2018) introduced a compression scheme in addition to a new exchange mode to reduce MIKEY_TICKET overhead.

Besides the proposed standard-based schemes, several approaches that aim to offload resource consuming operations to third entities have been proposed. Abdmeziem and Charoy (2018) highlighted the relevance of including controllers to support resource consuming primitives, while establishing group keys for collaborative systems. In addition, Saied and Olivereau (2012) introduced a collaborative approach for HIP. The idea is to take advantage of more powerful nodes in the neighborhood of a constrained node to carry heavy computations in a distributed way. Likewise, IKE session establishment delegation to the gateway has been proposed in Bonetto et al. (2012). Furthermore, Freeman et al. (2007) introduce a delegation procedure that enables a client to delegate certificate validation to a trusted server. While the precedent delegation approaches reduce the computational load at the constrained node, they break the end to end principle by requiring a third trusted party. Abdmeziem and Tandjaoui (2015) addressed the precedent issue by enhancing the existing schemes to ensure the end to end property.

Our approach combines the solutions from both research axes. In fact, it is based on the offloading of heavy asymmetric operations to third parties, and on a new compression scheme for MIKEY header. To the best of our knowledge, no prior similar work has been proposed for MIKEY applied to collaborative applications in the context of internet of things.

7 Conclusions and perspectives

We addressed the issue of setting secure communication channels between constrained entities and more powerful entities within a collaborative IoT context. We proposed a combined compressed and distributed MIKEY mode to mitigate both computational and communication costs. To reduce the size of the exchanged messages, we introduced a new header compression scheme that allows lightening the size of the header from 12 bytes to 3 bytes in the best compression case. In addition, to mitigate the computational overhead, heavy operations are offloaded to a set of dedicated third parties. Doing so, the constrained entities are only involved in the symmetric operations of the pre-shared mode. The public key mode is left to the unconstrained part of the network. As a result, the constrained entities are able to establish a secured channel with any remote entity without having established an initial shared knowledge. Our security evaluation involved a formal analysis of MIKEY security properties using Avispa tool. The results showed that our tailoring did not alter the security strength of MIKEY protocol. Moreover, our performance evaluation using energy models allowed us to highlight the obtained energy gains which increased with the increase of the compression rate of MIKEY header. Besides, the inclusion of additional third parties leads to a progressive inflation of energy consumption, hence a trade-off should be set between security strength and performances. As a future work, we plan an implementation on real test-beds to assess energy consumption performances under real conditions. In addition, we intend to investigate lightweight group key management protocols for collaborative IoT applications.

References

Abdelfadeel, K.Q., Cionca, V. and Pesch, D. (2017) 'LSCHC: layered static context header compression for lpwans', *Proceedings of the 12th Workshop on Challenged Networks, CHANTS '17*, pp.13–18, ACM, New York, NY, USA.

Abdmeziem, M. and Tandjaoui, D. (2015) 'An end-to-end secure key management protocol for e-health applications', *Computers & Electrical Engineering*, Special issue on 'Emerging Research in Internet of Things', Vol. 44, pp.184–197.

Abdmeziem, M., Tandjaoui, D. and Romdhani, I. (2018) 'Lightweighted and energy-aware MIKEY-ticket for e-health applications in the context of internet of things', *International Journal of Sensor Networks*, Vol. 26, No. 4, pp.227-242.

Abdmeziem, M.R. and Charoy, F. (2018) 'Fault-tolerant and scalable key management protocol for IoT-based collaborative groups', *Security and Privacy in Communication Networks: SecureComm 2017 International Workshops, ATCS and SePrIoT, Proceedings 13th*, 22–25 October 2017, Niagara Falls, ON, Canada, pp.320–338, Springer.

Abdmeziem, M.R., Tandjaoui, D. and Romdhani, I. (2016a) 'Architecting the internet of things: state of the art', *Robots and Sensor Clouds*, pp.55–75, Springer.

Abdmeziem, M.R., Tandjaoui, D. and Romdhani, I. (2016b) 'A new distributed MIKEY mode to secure e-health applications', *Proceedings of the International Conference on Internet of Things and Big Data – Volume 1: IoTBD*, pp.88–95.

Arkko, J., Lindholm, F., Naslund, M. and Norrman, K. (2004) *MIKEY: Multimedia Internet Keying*, RFC 3830, IETF.

AVISPA – A Tool for Automated Validation of Internet Security Protocols [online] http://www.avispa-project.org (accessed 17 March 2019).

Benouaret, K., Valliyur-Ramalingam, R. and Charoy, F. (2013) 'CrowdSC: building smart cities with large-scale citizen participation', *IEEE Internet Computing*, Vol. 17, No. 6, pp.57–63.

Bonetto, R., Bui, N., Lakkundi, V., Olivereau, A., Serbanati, A. and Rossi, M. (2012) 'Secure communication for smart IoT objects: protocol stacks, use cases and practical examples', *Proc. of IEEE WoWMoM*.

Cheng, T-F., Chen, Y-C., Song, Z-D., Huynh, N-T. and Lee, J-S. (2019) 'Secure session between iot device and cloud server based on elliptic curve cryptosystem', *International Journal of Information and Computer Security*, in press.

Chevalier, Y., Compagna, L., Cuellar, J., Drielsma, P.H., Mantovani, J., Modersheim, S. and Vigneron, L. (2004) 'A high level protocol specification language for industrial security sensitive protocols', *Proc. SAPS 04*, Austrian Computer Society.

De Meulenaer, G., Gosset, F., Standaert, F-X. and Pereira, O. (2008) 'On the energy cost of communication and cryptography in wireless sensor networks', *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2008. WIMOB'08*, pp.580–585, IEEE.

Dolev, D. and Yao, C. (1981) 'On the security of public key protocols', *FOCS*, pp.350–357, IEEE.

Dworkin, M. (2007) *Recommendation for Block Cipher Modes of Operation: The CCM Mode for Authentication and Confidentiality*, SP-800-38C, NIST, US Department of Commerce.

Freeman, T., Housley, R., Malpani, A., Cooper, D. and Polk, W. (2007) *Server-Based Certificate Validation Protocol (SCVP)*, RFC 5055, IETF.

Gnimpieba, Z.D.R., Nait-Sidi-Moh, A., Durand, D. and Fortin, J. (2015) 'Using internet of things technologies for a collaborative supply chain: application to tracking of pallets and containers', *Procedia Computer Science*, *The 10th International Conference on Future Networks and Communications (FNC 2015)/The 12th International Conference on Mobile Systems and Pervasive Computing (MobiSPC 2015) Affiliated Workshops*, Vol. 56, pp.550–557.

Hui, J. and Thubert, P. (2011) *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*, RFC 6282, IETF.

Hummen, R., Hiller, J., Henze, M. and Wehrle, K. (2013a) 'Slimfit – a HIP DEX compression layer for the IP-based internet of things', *WiMob*, pp.259–266, IEEE.

Hummen, R., Hiller, J., Wirtz, H., Henze, M., Shafagh, H. and Wehrle, K. (2013b) '6LoWPAN fragmentation attacks and metigation mechanisms', *Proc. 6th ACM Conf. Security Privacy Wireless Mobile Networks*, pp.55–66.

Jing, Q., Vasilakos, A.V., Wan, J., Lu, J. and Qiu, D. (2014) 'Security of the internet of things: perspectives and challenges', *Wireless Networks*, Vol. 20, No. 8, pp.2481–2501.

Kaps, J-P. and Sunar, B. (2006) 'Energy comparison of AES and SHA-1 for ubiquitous computing', *International Conference on Embedded and Ubiquitous Computing*, pp.372–381, Springer.

Khan, M.A. and Salah, K. (2017) 'IoT security: review, blockchain solutions, and open challenges', *Future Generation Computer Systems*, Vol. 82, pp.395-411.

Kouicem, D.E., Bouabdallah, A. and Lakhlef, H. (2018) 'Internet of things security: a top-down survey', *Computer Networks*, Vol. 141, pp.199-221.

Lin, J., Yu, W., Zhang, N., Yang, X., Zhang, H. and Zhao, W. (2017) 'A survey on internet of things: architecture, enabling technologies, security and privacy, and applications', *IEEE Internet of Things Journal*, Vol. 4, No. 5, pp.1125–1142.

Montenegro, G., Kushalnagar, N., Hui, J. and Culler, D. (2007) *Transmission of IPv6 Packets over IEEE 802.15.4 Networks*, RFC 4944, IETF.

Raza, S., Duquennoy, S., Chung, T., Yazar, D., Voigt, T. and Roedig, U. (2011) 'Securing communication in 6LoWPAN with compressed IPsec', *Proc. of IEEE DCOSS*.

Raza, S., Trabalza, D. and Voigt, T. (2012a) '6LowPAN compressed DTLS for CoAP', *Proc. of IEEE DCOSS*.

Raza, S., Voigt, T. and Jutvik, V. (2012b) 'Lightweight IKEv2: a key management solution for both compressed IPsec and IEEE 802.15.4 security', *IETF/IAB Workshop on Smart Object Security*.

Reed, S. and Solomon, G. (1960) 'Polynomial codes over certain finite fields', *Journal of the Society for Industrial and Applied Mathematics*, Vol. 8, No. 2, pp.300–304.

Roman, R., Alcaraz, C., Lopez, J. and Sklavos, N. (2011) 'Key management systems for sensor networks in the context of internet of things', *Computers and Electric Engineering*, Vol. 37, No. 2, pp.147–159.

Sahraoui, S. and Bilami, A. (2015) 'Efficient HIP-based approach to ensure lightweight end-to-end security in the internet of things', *Computer Networks*, Vol. 91, pp.26–45.

Saied, Y.B. and Olivereau, A. (2012) 'HIP tiny exchange (TEX): a distributed key exchange scheme for HIP-based internet of things', *Proc. of ComNet*.

Sicari, S., Rizzardi, A., Miorandi, D. and Coen-Porisini, A. (2016) 'Internet of things: security in the keys', *Proceedings of the 12th ACM Symposium on QoS and Security for Wireless and Mobile Networks*, pp.129–133, ACM.

Tolone, W., Ahn, G-J., Pai, T. and Hong, S-P. (2005) 'Access control in collaborative systems', *ACM Computing Surveys (CSUR)*, Vol. 37, No. 1, pp.29–41.

Tsiftes, N. and Dunkels, A. (2011) 'A database in every sensor', *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*, pp.316–332.

Wander, A., Gura, N., Eberle, H., Gupta, V. and Shantz, S. (2005) 'Energy analysis of public-key cryptography for wireless sensor networks', *Third IEEE International Conference on Pervasive Computing and Communications*.

Notes

1 In Figure 9, notice that for a better clarity of results presentation, we divided the energy consumption of asymmetric primitives by a factor of 10.