Distributed and Compressed MIKEY Mode to Secure End-to-End Communications in the Internet of Things

Mohammed Riyadh Abdmeziem
LSI, USTHB:
University of Sciences
and Technology Houari Boumedienne
32, El Alia, Bab Ezzouar
Algiers, Algeria
rabdmeziem@usthb.dz

Djamel Tandjaoui
CERIST:
Center for Research on Scientific
and Technical Information
03, Rue des freres Aissou, Ben Aknoun
Algiers, Algeria
dtandjaoui@mail.cerist.dz

Imed Romdhani
School of Computing,
Edinburgh Napier University
10, Colinton Road EH10 5DT, Edinburgh
United Kingdom (UK)
I.Romdhani@napier.ac.uk

***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 distributed and compressed MIKEY mode for the Internet of Things. Indeed, relying on a cooperative approach, a set of third parties is used to discharge the constrained nodes from heavy computational operations. Doing so, the 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. Preliminary results show that our proposed mode is energy preserving whereas its security properties are preserved untouched.**

Keywords*—Internet of Things (IoT), E-health, MIKEY, Key management protocols, Security.

I. 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 [1]. It is considered as one of the most important communication development in recent years. It makes our everyday objects (e.g. health sensors, industrial equipements, vehicles, clothes, etc.) connected to each other and to the Internet [2]. Among the emerging IoT applications, e-health and telecare are gaining more and more attention. In fact, population ageing and the increase of survival chances from disabling accidents lead to an increased demand for continuous health care and monitoring [3].

Compared to other IoT applications, e-health applications are more vulnerable to attacks due to the high sensitivity of the generated data [4]. These data are private in nature, and any security vulnerability regarding the confidentiality would seriously repulse patients from adopting e-health applications, and therefore increase governments health spending cost. Securing data communications for e-health applications passes inevitably through reliable and robust key management protocols. These protocols are in charge of delivering secure credentials to the different communicating entities. These credentials are used to make sure that only authorized entities can access and modify data.

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) [5]. The flexibility of MIKEY allows the designers to leverage upon several modes according to the specificities of the network scenario. Thus, MIKEY seems to be the adequate protocol that can be extended to ensure secure communications in the 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.

In this paper, we extend our two previous approaches [6] [7] to propose a new standard-based distributed and compressed key management scheme. In fact, we design a new hybrid mode for MIKEY protocol that mitigates both the computational and communication costs. Firstly, 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 to mitigate the

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

TABLE I: Terminology table

disadvantages of both Pre-shared key mode and the Public Key mode while benefiting from their advantages.

Secondly, we propose a new 6LoWPAN (IPv6 over Low power Wireless Personal Area Networks) header compression scheme to reduce the communication cost. Our scheme is intended to save energy and avoid 6LoWPAN fragmentation that may occur when a datagram size exceeds the link layer MTU [1]. Indeed, fragmentation is undesirable, as 6LoWPAN is vulnerable to fragmentation attacks [8]. As a first assessment of our approach, we conducted a theoretical analysis of the security properties. Furthermore, we formally validated the analysis using Avispa tool [9]. The obtained results showed that our approach keeps the security properties safe while being energy efficient.

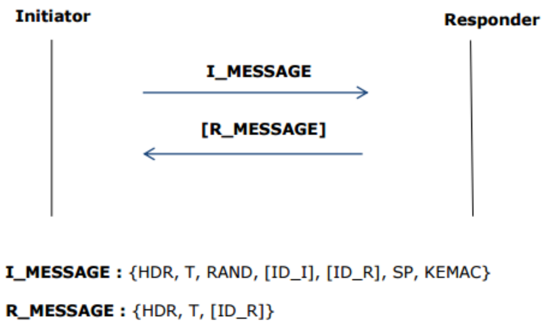
The remaining of the paper is organized as follows. Section 2 provides the required background for a clear comprehension of the proposed approach. In section 3, we introduce our new hybrid MIKEY mode. First, we present our network model and assumptions. Then, we detail the proposed approach. In section 4, we analyze the security properties of our proposed mode. Existing security solutions in the literature are surveyed in section 5. Section 6 concludes the paper and sets our future directions.

II. Background

In this section, we provide an overview of the features of MIKEY protocol [5] while focusing on the adaptability of its different modes to the constrained environment of e-health applications. In addition, we briefly present the concepts used throughout the remaining of the paper.

A. MIKEY overview

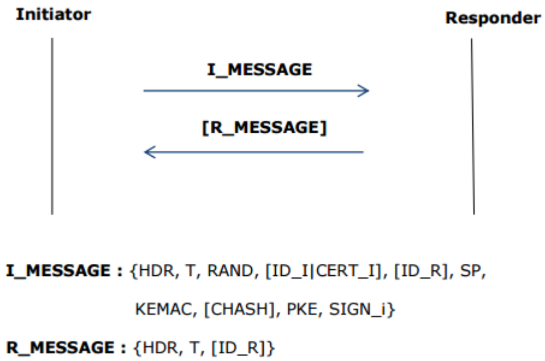
MIKEY considers two entities that aim to establish a shared secret. One of the two entities assumes the *Initiator role*, whereas the second one assumes the *Responder* role. The key



I_MESSAGE : {HDR, T, RAND, [ID_I], [ID_R], SP, KEMAC}

R_MESSAGE : {HDR, T, [ID_R]}

Fig. 1: Pre-shared key mode signaling flow



I_MESSAGE : {HDR, T, RAND, [ID_I|CERT_I], [ID_R], SP, KEMAC, [CHASH], PKE, SIGN_i}

R_MESSAGE : {HDR, T, [ID_R]}

Fig. 2: Public key mode signaling flow

distribution modes are defined as follows (the different used notations are described in Table I):

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 $auth\_key$ (Fig. 1).

In the pre-shared key mode, only symmetric operations are involved. Hence, this mode fits well 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. Indeed, a pre-establishment phase is required, where 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.

[1] Maximum Transmission Unit of the IEEE 802.15.4 protocol

According to [5], the mandatory asymmetric primitive to implement is RSA [10]. 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 (Fig. 2).

The Public key mode is based on asymmetric primitives (i.e. RSA). 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.

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 e-health scenario, this mode is ruled out.

B. Common Header Format (HDR)

The Common Header payload contains information about the different exchanged messages. It is always present as the first payload in each message. In the following, we present a succinct description of each field contained in the MIKEY header. We refer to RFC3830 [5] 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.

C. 6LoWPAN Adaptation Layer

The 6LoWPAN standard defined in [11] aims to transfer IPv6 packets to 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 only 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 the reader to RFC 6282 [11].

III. Contributions

In this section, we introduce a new hybrid mode for MIKEY protocol. Firstly, we present our e-health network architecture. Secondly, we define a set of assumptions before detailing our contributions.

A. Network architecture and assumptions

We consider an end-to-end communication channel between smart objects (i.e. sensor nodes) and any remote server. This choice is motivated by the high sensitivity of gathered data in e-health applications. Hence, key management protocols are required between the two entities to secure their communications. These protocols have to deal with the resources capabilities of the involved entities, along with the fact that no prior knowledge is established between them.

IP-enabled smart objects are in charge of sensing health related data (e.g. blood pressure, blood glucose level, temperature level, etc.). They are planted in the human body. Gathered data is transmitted to remote entities that are in charge of the processing and analysis. In our approach, we consider four main elements: the mobile and contextual sensors, the third parties, the remote server and the certification authority. (Fig. 3).

- ***Mobile and contextual sensors:*** the sensors are planted in, on, or around a human body to collect health-related data (e.g. blood pressure, blood glucose level, temperature level, etc.).

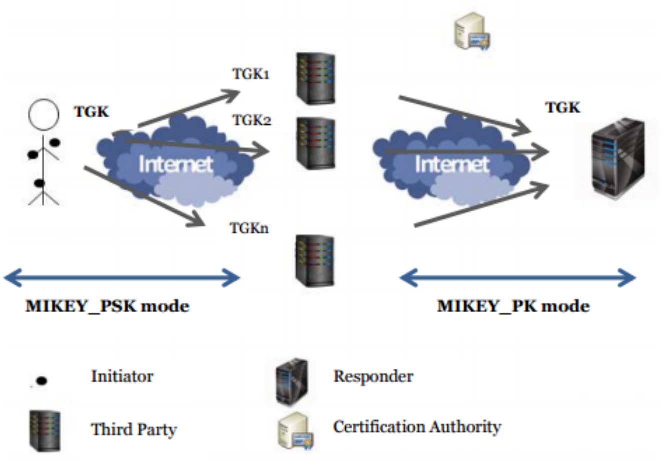


Fig. 3: MIKEY hybrid mode: network architecture

- ***Third Party:*** Compared to the standard MIKEY modes, the third parties represent an additional component in our proposed hybrid mode . A third party could be any entity that is able to perform high consuming computations.

- ***Remote server:*** the remote server receives the gathered data for further processing. A remote server could be used by caregiver services in order to take appropriate decisions according to patient's data.
- ***Certification authority:*** the certification authority is required to guarantee authentication between the third parties and the remote server by delivering valid and authenticated certificates.

The network is thus heterogeneous combining nodes with various capabilities both in terms of computing power and energy resources. Smart objects have limited computational power, memory and energy resources. They are unable to perform public key cryptographic operations. However, the third parties and the remote server are equipped with high energy, computing power and storage capabilities. They can take the form of a 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 smart object (also designated as constrained node), while *the Responder* is mapped with the remote entity, which can be located in hospitals and automatically trigger an exchange to check on patient's vital signs.

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

- Sensor nodes are able to perform symmetric encryption. Both third parties and the remote server 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 to the third parties and to the remote server.
- Each sensor node is able to keep a list of remote third parties. This list is pre-established during the initialization phase.
- Each sensor node shares a PSK with each third party.

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

TABLE II: Gained space through the proposed MIKEY Common Header compression

B. *Reducing MIKEY communication overhead (Compression)*

In this section, we describe our proposed 6LoWPAN header compression scheme for MIKEY. Our compression is based on the fact 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 removed. As explained in section II.C, 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 [12] 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.

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 II and Fig. 4).

- ***Version (V):*** if 0, the version is the default and latest MIKEY version defined in [5] and the field is skipped. If future versions are defined, the bit is set

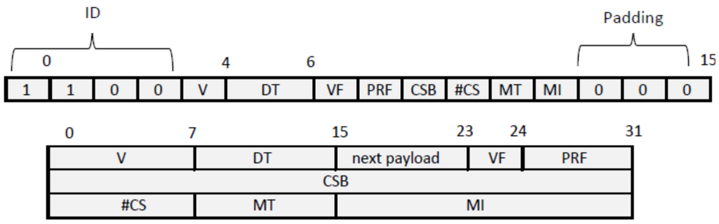


Fig. 4: Our 6LoWPAN-NHC-HDR encoding compared to the basic MIKEY header

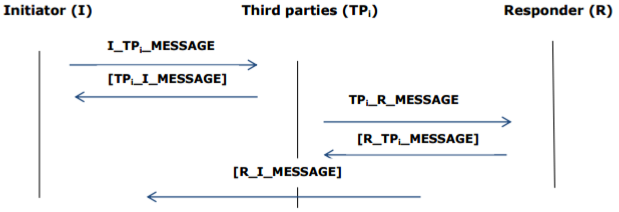


Fig. 5: Illustration of the different message exchanges of our proposed mode

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 III.C), 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 [5] 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 [5] 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 [5]. 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 RFC6282 [13] (NHC size is defined as 2 bytes long).

C. Reducing MIKEY computation overhead (Distribution)

In our hybrid mode, the network is divided into two segments. The first segment is defined by the communication channel linking the constrained nodes to the third parties. This segment involves the constrained part of our network model. Hence, we propose to consider using the Pre-shared key mode. The second segment is defined by the communication channel linking the third parties to the remote server. This segment does not suffer from resources constraints, thus, we propose to consider using the Public key mode.

After an initialization phase where each constrained node is pre-loaded with a set of third parties identities, along with the different PSK, our proposed new MIKEY mode proceeds with successive messages. Table. I summarizes the notations used, and Fig. 5 illustrates the signaling flow. To remain standard compliant, the messages header, along with various message parameters are kept unchanged (RFC 3830 [5]). 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,

$...TGK_n$. Each part is sent to the appropriate TP_i in *I_TPi_MESSAGE*. The message is replay protected with timestamps, encrypted and authenticated using the pre_shared PSK. The general structure of the message is as follows.

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

Because wireless connection is the main media in e-health applications, and in IoT in general, I applies an error redundancy scheme to the generated TGK. The aim is to enable R retrieving the secret without requiring the reception of all the packets, in case where some of them were lost during the transmission process. For instance, the widely used Reed-Solomon scheme can be applied [14].

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

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

- ***TPi_R_MESSAGE:*** after having properly authenticated the received *I_TPi_MESSAGE*, TP_i randomly generates an envelope key. This key is used to encrypt and authenticate the received TGK_i part, which is included in *TPi_R_MESSAGE*. The envelope key is encrypted with the public key of R and included in the message. In addition, TP_i's signature that covers all the fields of the message is also included. The message is then sent to R. The structure of the message is as follows.

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

- ***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.

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

- ***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[5]. 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 sensors and remote unconstrained servers. Our hybrid mode takes advantage of both the Pre-shared and Public-key modes, while limiting their disadvantages.

IV. Security Analysis

A. Key exchange properties

In this section, we briefly analyze the security features of our proposed mode based on the properties presented in [15]. For the following discussion, we consider our communication channel split into two segments: Seg1) from I to the TP_i and Seg2) from the TP_i to R (see Fig. 3)

Confidentiality: regarding Seg1, the exchanged messages between I and the different TP_i are encrypted using the corresponding PSK_i. Based on RFC 3830 [5], we advocate the use of AES-CCM mode that defines AES-CBC for MAC generation and AES-CTR for encryption [16]. 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. According to RFC 3830 [5], RSA is used as a cryptographic primitive [10]. 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. Hence, the exchanged data are guaranteed to remain genuine. This property ensures that the 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 initialization 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 R establish a secure channel in an online mode taking advantage from the asymmetric primitives. As a consequence, upon an initial distribution in Seg1, our hybrid mode can be run without any external intervention allowing automatic updates.

Overhead: in our hybrid mode, the constrained entities are only involved in symmetric operations, which are much less resource consuming than asymmetric ones. 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 hybrid mode does not assume the third parties to be trusted.

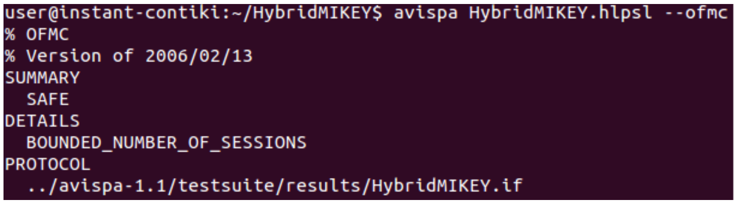
Extensibility and scalability: in an e-health scenario, new sensors can be integrated at any time. We can easily imagine a physician prescribing the implantation of a new sensor for various medical purposes. Our protocol requires an initialization phase where the sensor (i.e. I) is set with a list of 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 R. After the initialization phase, the joining sensor is ready to establish an end to end secure channel with any remote entity.

Storage: due to recent hardware advances in flash memory, smart objects provide considerable amounts of storage space. This space is used in our hybrid mode 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. Thus, storage space is not considered as an issue in our protocol deployment.

B. Formal validation

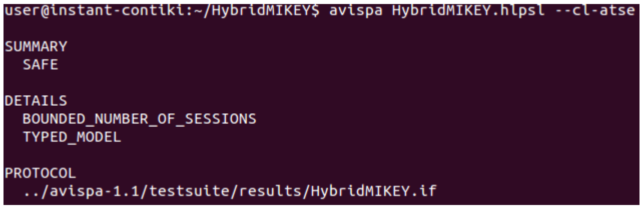
To prove that our protocol does not violate the required security properties, in particular, confidentiality, authentication, delivery proof and replay protection, we carried out an analysis using Avispa tool [9]. AVISPA (Automated Validation of Internet Security Protocol and Applications) is a state-of-the-art 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 [17] 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.

Protocol models in Avispa are written in a role-based language called High Level Protocol Specification Language, or HLPSL [18]. 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 analyzed protocol are specified in the *goal section* before launching the analysis. Besides, Avispa uses several different automatic protocol analysis techniques to validate the analyzed protocol against the specified security goals such as the on-the-fly model-checker (OFMC), and the constraint-logic based attack searcher (CL-AtSe).



```
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
```

Fig. 6: Avispa output (OFMC)



```
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
```

Fig. 7: Avispa output ($CL - AtSe$)

In our modeling, we have first specified a *basic role* to describe the actions of the different entities involved. Then, we have 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 anayzed against the Dolev-Yao intruder model using the OFMC, 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 (Fig. 6) , and $CL - AtSe$ (Fig. 7). Based on the obtained results, we can affirm that our proposed distributed mode is safe with respect to the specified security goals.

V. Related Work

In our literature review, we distinguish two main research axes. The first one is focused on compression schemes applied on standard based protocols, while the second one is focused on the approaches based on the offloading of heavy computational operations to third parties. Numerous energy aware approaches have been introduced for the IP-based IoT. In [13] and [11], the compression of IPV6 headers, extension headers along with UDP headers has been standardized through 6LoWPAN. Authors in [19] presented 6LoWPAN compressions for IPsec payload headers (AH and ESP). In [20], an IKE compression scheme has also been proposed providing a lightweight automatic way to establish security associations for IPsec. Likewise, header compression layer for DTLS, HIP DEX, and HIP BEX was respectively introduced in [12], [21], and [22]. Furthermore, in [6], authors 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. Authors in [23] 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 have been proposed in [24]. Furthermore, authors in [25] 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. Authors, in [7], addressed the precedent issue by enhancing the existing schemes to ensure the end to end property.

Our solution combines the approaches from both axes. In fact, it is based on the offloading of heavy asymmetric operations to third parties, while introducing a new compression scheme for a standard based protocol (i.e. MIKEY). To the best of our knowledge, no prior similar work has been proposed for MIKEY protocol applied to e-health applications in the context of Internet of Things.

VI. Conclusions and perspectives

We addressed the problematic of establishing secure communication channels in the constrained environment of e-health applications. In fact, we introduced a new MIKEY mode that combines distribution and compression to reduce both computational and communication costs. In our mode, heavy operations are offloaded to dedicated powerful 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. Indeed, through our hybrid mode, we benefit from the advantages of both pre-shared mode (resource preservation) and public key mode (scalability), while mitigating their disadvantages. Moreover, we proposed a new header compression scheme to reduce the size of messages from 12 Bytes to 3 Bytes in the best compression case. The first preliminary results show that our mode is secure, and resource preserving at the same time. In the future, we plan an implementation on real test-beds to assess its energy consumption performances under real conditions.

References

[1] L. Atzori, A. Iera, and G. Morabito, "The internet of things: A survey," *Computer Networks*, pp. 2787–2805, May 2010.

[2] M. R. Abdmeziem, D. Tandjaoui, and I. Romdhani, "Architecting the internet of things: State of the art," in *Robots and Sensor Clouds*. Springer, 2016, pp. 55–75.

[3] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, "The internet of things for ambient assisted living," in *Information Technology: New Generations (ITNG)*, April 2010, pp. 804–809.

[4] M. Li and W. Lou, "data security and privacy in wireless body area networks," *Wireless Technologies for E-healthcare*, February 2010.

[5] J. Arkko, F. Lindholm, M. Naslund, and K. Norrman, "Mikey: Multimedia internet keying," *RFC 3830, IETF*, 2004.

[6] M. R. Abdmeziem and D. Tandjaoui, "Tailoring mikey-ticket to e-health applications in the context of internet of things," in *International Conference on Advanced Networking, Distributed Systems and Applications (Short Papers)*, June 2014, pp. 72–77.

[7] M. Abdmeziem and D. Tandjaoui, "An end-to-end secure key management protocol for e-health applications," *Computers & Electrical Engineering*, 2015.

[8] R. Hummen, J. Hiller, H. Wirtz, M. Henze, H. Shafagh, and K. Wehrle, "6lowpan fragmentation attacks and metigation mechanisms," *Proc. 6th ACM Conf. Security Privacy Wireless Mobile Networks*, pp. 55–66, Apr 2013.

[9] "Avispa a tool for automated validation of internet security protocols," *http://www.avispa-project.org*.

[10] R. L. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems," *Communications of the ACM*, vol. 21, no. 2, pp. 120–126, 1978.

[11] J. Hui and P. Thubert, "Compression format for ipv6 datagrams over ieee 802.15.4-based networks," *RFC 6282, IETF*, 2011.

[12] S. Raza, D. Trabalza, and T. Voigt, "6lowpan compressed dtls for coap," *in Proc. of IEEE DCOSS*, 2012.

[13] G. Montenegro, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of ipv6 packets over ieee 802.15.4 networks," *RFC 4944, IETF*, 2007.

[14] S. Reed and G. Solomon, "Polynomial codes over certain finite fields," *Journal of the Society for Industrial and Applied Mathematics*, vol. 8, no. 2, pp. 300–304, 1960.

[15] R. Roman, C. Alcaraz, J. Lopez, and N. Sklavos, "key management systems for sensor networks in the context of internet of things," *Computers and Electric Engineering*, vol. 37, pp. 147–159, 2011.

[16] M. Dworkin, "Recommendation for block cipher modes of operation: The ccm mode for authentication and confidentiality," *SP-800-38c, NIST, US department of commerce*, 2007.

[17] D. Dolev and C. Yao, "On the security of public key protocols," *FOCS, IEEE, 1981*, p. 350357, 1981.

[18] Y. Chevalier, L. Compagna, J. Cuellar, P. H. Drielsma, J. Mantovani, and a. L. V. S. Modersheim, "A high level protocol specification language for industrial security sensitive protocols," *Proc. SAPS 04. Austrian Computer Society, 2004*, 2004.

[19] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig, "Securing communication in 6lowpan with compressed ipsec," *in Proc. of IEEE DCOSS*, 2011.

[20] S. Raza, T. Voigt, and V. Jutvik, "Lightweight ikev2: A key management solution for both compressed ipsec and ieee 802.15.4 security," *IETF/IAB workshop on Smart Object Security*, 2012.

[21] R. Hummen, J. Hiller, M. Henze, and K. Wehrle, "Slimfit a hip dex compression layer for the ip-based internet of things," *WiMob, IEEE*, pp. 259–266, 2013.

[22] S. Sahraoui and A. Bilami, "Efficient hip-based approach to ensure lightweight end-to-end security in the internet of things," *Computer Networks*, vol. 91, pp. 26–45, 2015.

[23] Y. B. Saied and A. Olivereau, "Hip tiny exchange (tex): A distributed key exchange scheme for hip-based internet of things," *in Proc. of ComNet*, 2012.

[24] R. Bonetto, N. Bui, V. Lakkundi, A. Olivereau, A. Serbanati, and M. Rossi, "Secure communication for smart iot objects: Protocol stacks, use cases and practical examples," *In Proc. of IEEE WoWMoM*, 2012.

[25] T. Freeman, R. Housley, A. Malpani, D. Cooper, and W. Polk, "Server-based certificate validation protocol(scvp)," *RFC 5055, IETF*, 2007.