A New Distributed MIKEY Mode to Secure e-Health Applications

Mohammed Riyadh Abdmeziem[1], Djamel Tandjaoui[2] and Imed Romdhani[3]
[1]*LSI, USTHB: Bab Ezzouar, Algiers, Algeria*
[2]*Security Division, CERIST: Ben Aknoun, Algiers, Algeria*
[3]*School of Computing, Edinburgh Napier University, Edinburgh, U.K.*

Keywords: Internet of Things (IoT), e-Health, Key Management Protocols, MIKEY, Security, Privacy, Cooperation.

Abstract: Securing e-health applications in the context of Internet of Things (IoT) is challenging. Indeed, resources scarcity in such environment hinders the implementation of existing standard based protocols. Among these protocols, MIKEY (Multimedia Internet KEYing) aims at establishing security credentials between two communicating entities. However, the existing MIKEY modes fail to meet IoT specificities. In particular, the pre-shared key mode is energy efficient, but suffers from severe scalability issues. On the other hand, asymmetric modes such as the public key mode are scalable, but are highly resource consuming. To address this issue, we combine two previously proposed approaches to introduce a new distributed MIKEY mode. 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 the network, while the public key mode is used in the unconstrained part of the network. Preliminary results show that our proposed mode is energy preserving whereas its security properties are kept safe.

1 INTRODUCTION

Internet of Things (IoT) is based on the pervasive presence around us 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 (Atzori et al., 2010). 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, vehicules, clothes, etc.) connected to each other and to the Internet (Abdmeziem et al., 2016). Among the different applications of IoT, e-health is gaining more and more attention in the IoT world. In fact, population ageing and the increase of survival chances from disabling accidents lead to an increased demand for continuous health care and monitoring (Dohr et al., 2010).

Compared to other IoT applications, e-health applications are more vulnerable to attacks due to the high sensitivity of the generated data (Li and Lou, 2010). This data is private in nature, and any security vulnerability regarding the confidentiality would seriously repulse patients from adopting e-health applications. For example, personal health information such as an early stage of pregnancy or details of certain medical conditions must be kept confidential. The leaked data can be used for illegal activities. In addition, any malicious alteration of health records would engender dramatic consequences, as it could trigger wrong medical prescriptions, or delay emergency interventions. Securing data communications for e-health applications passes inevitably through key management protocols. They are in charge of delivering security credentials to the different involved entities. These credentials are used to make sure that only authorized entities can access and modify data. This is particularly relevant in an e-health scenario considering its sensitivity.

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. Thus, MIKEY seems to be the adequate protocol that can

Abdmeziem, M., Tandjaoui, D. and Romdhani, I.
A New Distributed MIKEY Mode to Secure e-Health Applications.
DOI: 10.5220/0005945300880095
In *Proceedings of the International Conference on Internet of Things and Big Data (IoTBD 2016)*, pages 88-95
ISBN: 978-989-758-183-0

be extended to ensure secure communications in IoT. 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 introduce an extension of our two previous approches (Abdmeziem and Tandjaoui, 2014) (Abdmeziem and Tandjaoui, 2015) to propose a new standard-based cooperative key management scheme. In fact, we design a new distributed hybrid mode for MIKEY protocol combining the Pre-shared key mode with the Public key mode. 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 overhead 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 hybrid mode allows us to mitigate the disadvantages of both Pre-shared key, and the Public Key modes while benefiting from their advantages. Indeed, the constrained nodes do not suffer from the scalability issue, as they can establish a secret with any remote entity without having a previously shared knowledge. In the same time, they are only involved in simple operations, which are compliant with e-health applications limitations.

As a first assessment of our approach, we conducted a theoretical analysis of its security properties. Furthermore, we formally validated the analysis using Avispa tool (Moedersheim and Drielsma, 2003). 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 a general overview on MIKEY protocol. In section 3, we introduce our new MIKEY mode. Firstly, we present our network architecture. Then, we set our assumptions, before detailing the protocol's functioning. In section 4, we analyze the security properties of our proposed mode. Existing security approaches are reviewed in section 5. Section 6 concludes the paper and sets our future directions.

2 BACKGROUND

In the following, we provide the necessary background on MIKEY's functioning (Arkko et al., 2004).

Table 1: Terminology table.

Notation	Description
I	Initiator
R	Responder
$\{data\}_k$	Data encrypted with key k
$[data]$	Optional data
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

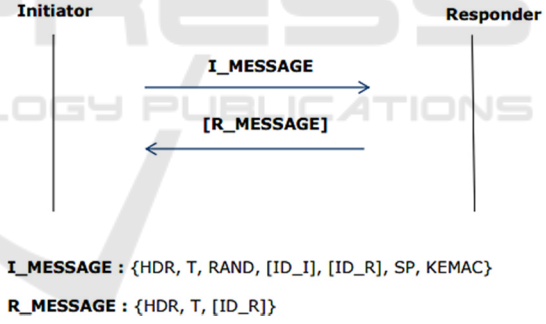

Figure 1: Pre-shared key mode signaling flow.

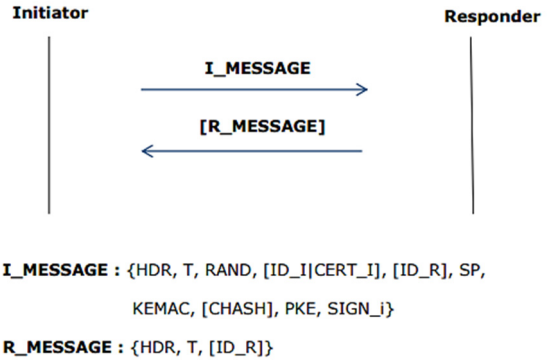

Figure 2: Public key mode signaling flow.

MIKEY considers two entities that aim to establish a shared secret. One of the two entities assumes the *Ini- tiator* role, whereas the second one assumes the *Re- sponder* role. The key distribution modes are defined as follows (the different used 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* (Fig. 1).

In the pre-shared key mode, only symmetric operations are involved. Hence, this mode fits well with the IoT constrained environement, as it can be run with limited energy and power resources. Nevertheless, this mode suffers from a severe scalability issue. In fact, 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 genrated 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. According to (Arkko et al., 2004), the mandatory asymmetric primitive to implement is RSA (Rivest et al., 1978). In case where the *Responder* owns several public keys, the *Initiator* specifies the used key in the optional 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 latter 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 latter is mainly based on the Diffie-Hellman key exchange protocol. This mode has a higher computational and communication overhead compared to the public key and the Pre-shared modes. Due to its inadequacy with our constrained e-health scenario, this mode is ruled out.

3 CONTRIBUTION

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

3.1 NeTwork Architecture

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 Sensor (Smart Object):* 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.).
- *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 Entity:* the remote entity 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.

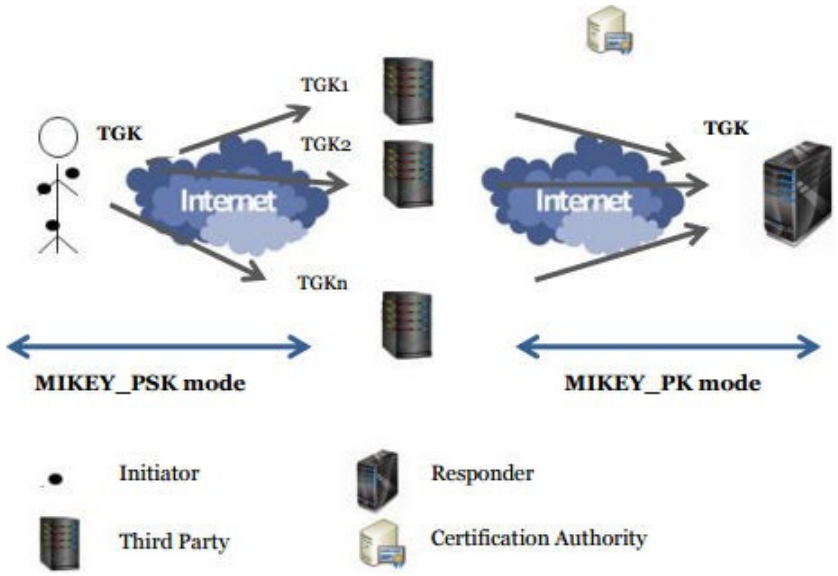

Figure 3: Mikey hybrid mode: Network architecture.

- *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 equiped 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 (Chang and Ramachandran, 2016).

The mapping with MIKEY concepts is defined as follows.

- *The Initiator role* is mapped with the smart object (also designated as constrained node)
- *The Responder role* is mapped with the remote entity. This latter can be set in hospitals automatically triggering an exchange in order to check on patient's vital signs.

3.2 Assumptions

Before presenting the details of our protocol, 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 an initialization phase.
- Each sensor node shares a PSK with each third party.

3.3 Message Exchanges

In our proposed 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 of MIKEY. 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 of MIKEY.

After an initialization phase where each constrained node is pre-loaded with the identities of a set of third parties, along with the different *PSK*, our proposed MIKEY mode proceeds with successive messages. Table. 1 summarizes the notations used, and Fig. 4 illustrates the signaling flow. To remain standard compliant, the messages headers, 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. Each part is sent to the appropriate TP_i in *I_TPi_MESSAGE*. The message is replay pro- tected 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 (Reed and Solomon, 1960).
- *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 lat- ter 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 RFC3830(Arkko et al., 2004). Both *I* and *R* are then able to derive state connec- tion 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 proposed mode takes advantage of both the Pre-shared and Public-key modes, while limiting their disadvan- tages.

4 SECURITY ANALYSIS

4.1 Key Exchange Properties

In this section, we briefly analyze the security features of our proposed mode based on the proprieties presented in (Roman et al., 2011). 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 (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 coprocessor, which would help to decrease the overhead. Regarding Seg2, communications are secured using Public Key Encryption. According to RFC 3830 (Arkko et al., 2004), RSA is used as a cryptographic primitive (Rivest et al., 1978). 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. Thus, the exchanged data is 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

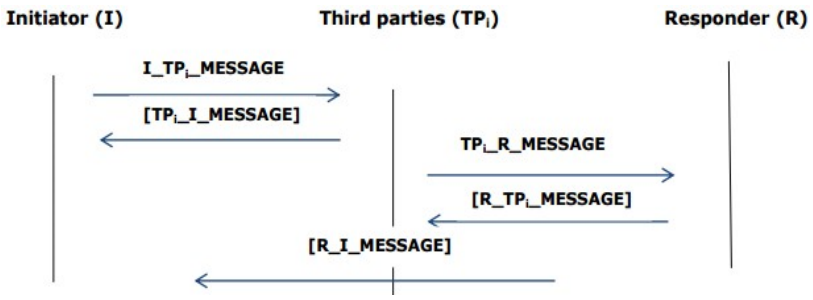

Figure 4: Distributed MIKEY mode: message exchanges.

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 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. 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 mode 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 (Tsiftes and Dunkels, 2011). 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.

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 (Moedersheim and Drielsma, 2003). 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 (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.

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 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).

In our modeling, we first specified a *basic role* to describe the actions of the different involved en-

```
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
```

Figure 5: 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
```

Figure 6: Avispa output ($CL-AtSe$).

tities. Then, we specified how the participants interact with each other in a *composed role*. The different roles have been implemented 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 have been indicated in reports for each backend model produced by Avispa tool. They show that our protocol is "SAFE" against OFMC (Fig. 5), and $CL-AtSe$ (Fig. 6). Based on the obtained results, we can affirm that our distributed hybrid mode is safe with respect to the specified security goals.

5 RELATED WORK

In our literature review, we distinguish two main research axes. The first one is focused on the security approaches designed upon 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 (Hui and Thubert, 2011), the compression of IPV6 headers, extension headers along with UDP headers has been standardized through 6LoWPAN. Authors in (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 and HIP DEX have been introduced in (Raza et al., 2012a), (Hummen et al., 2013), and (Sahraoui and Bilami, 2015). Furthermore, in (Abdmeziem and Tandjaoui, 2014), 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 (Saied and Olivereau, 2012) introduced collaboration 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 a gateway have been proposed in (Bonetto et al., 2012). Furthermore, authors in (Freeman et al., 2007) introduced 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 (Abdmeziem and Tandjaoui, 2015), addressed the precedent issue by enhancing the existing schemes to ensure the end to end property.

The proposed approach in this paper can be positioned in both axes. In fact, it is based on the offloading of heavy asymmetric operations to third parties, while being implemented through a standard based protocol (i.e. MIKEY).

6 CONCLUSIONS

We addressed the problematic of establishing secured communication channels in the constrained environment of e-health applications. In fact, we introduced a new MIKEY mode that combines the pre-shared key mode with the public key mode. In this 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 distributed hybrid mode, we benefit from the advantages of both pre-shared mode (resource preservation) and public key mode (scalability), while mitigating their disadvantages. 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

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

Abdmeziem, M. R. and Tandjaoui, D. (2014). Tailoring mikey-ticket to e-health applications in the context of internet of things. In *International Conference on Advanced Networking, Distributed Systems and Applications*, pages 72–77.

Abdmeziem, M. R., Tandjaoui, D., and Romdhani, I. (2016). Architecting the internet of things: State of the art. In *Robots and Sensor Clouds*, pages 55–75. Springer International Publishing.

Arkko, J., Lindholm, F., Naslund, M., and Norrman, K. (2004). Mikey: Multimedia internet keying. *RFC 3830, IETF*.

Atzori, L., Iera, A., and Morabito, G. (2010). The internet of things: A survey. *Computer Networks*, pages 2787–2805.

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. *In Proc. of IEEE WoWMoM*.

Chang, V. and Ramachandran, M. (2016). Towards achieving data security with the cloud computing adoption framework. *IEEE Transactions on Services Computing*, 9(1):138–151.

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

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

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

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*.

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. (2013). Slimfit a hip dex compression layer for the ip-based internet of things. *WiMob, IEEE*, pages 259–266.

Li, M. and Lou, W. (2010). Data security and privacy in wireless body area networks. *Wireless Technologies for E-healthcare*.

Moedersheim, S. and Drielsma, P. (2003). Avispa project deliverable d6.2: Specification of the problems in the high-level specification language. http://www.avispa-project.org.

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

Raza, S., Trabalza, D., and Voigt, T. (2012a). 6lowpan compressed dtls for coap. *in 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*, 8(2):300–304.

Rivest, R. L., Shamir, A., and Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126.

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*, 37:147–159.

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

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

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