Conference Paper

Tightening up security in low power deterministic networks

Walter Tiberti

Bruno Vieira

Harrison Kurunathan*

Ricardo Severino*

Eduardo Tovar*

*CISTER Research Centre

CISTER-TR-200312

2020

Tightening up security in low power deterministic networks

Walter Tiberti, Bruno Vieira, Harrison Kurunathan*, Ricardo Severino*, Eduardo Tovar*

*CISTER Research Centre

Polytechnic Institute of Porto (ISEP P.Porto)

Rua Dr. António Bernardino de Almeida, 431

4200-072 Porto

Portugal

Tel.: +351.22.8340509, Fax: +351.22.8321159

E-mail: walter.tiberti@graduate.univaq.it, bffbv@isep.ipp.pt, hhkur@isep.ipp.pt, sev@isep.ipp.pt, emt@isep.ipp.pt

https://www.cister-labs.pt

Abstract

The unprecedented pervasiveness of IoT systems is pushing this technology into increasingly stringent domains. Such application scenarios become even more challenging due to the demand for encompassing the interplay between safety and security. The IEEE 802.15.4 DSME MAC behavior aims at addressing such systems by providing additional deterministic, synchronous multi-channel access support. However, despite the several improvements over the previous versions of the protocol, the standard lacks a complete solution to secure communications. In this front, we propose the integration of TAKS, an hybrid cryptography scheme, over a standard DSME network. In this paper, we describe the system architecture for integrating TAKS into DSME with minimum impact to the standard, and we venture into analysing the overhead of having such security solution over application delay and throughput. After a performance analysis, we learn that it is possible to achieve a minor impact of 1\% to 14\% on top of the expected network delay, depending on the platform used, while still guaranteeing strong security support over the DSME network.

Tightening up security in low power deterministic networks

Walter Tiberti[§], Bruno Vieira[†], Harrison Kurunathan[†], Ricardo Severino[†], Eduardo Tovar[†]
[§] DEWS/University of L'Aquila, Italy and [†] CISTER/ISEP-IPP, Porto, Portugal
Email: walter.tiberti@univaq.it, (bffbv, hhkur, rarss, emt)@isep.ipp.pt

***Abstract*—The unprecedented pervasiveness of IoT systems is pushing this technology into increasingly stringent domains. Such application scenarios become even more challenging due to the demand for encompassing the interplay between safety and security. The IEEE 802.15.4 DSME MAC behavior aims at addressing such systems by providing additional deterministic, synchronous multi-channel access support. However, despite the several improvements over the previous versions of the protocol, the standard lacks a complete solution to secure communications. In this front, we propose the integration of TAKS, an hybrid cryptography scheme, over a standard DSME network. In this paper, we describe the system architecture for integrating TAKS into DSME with minimum impact to the standard, and we venture into analysing the overhead of having such security solution over application delay and throughput. After a performance analysis, we learn that it is possible to achieve a minor impact of 1% to 14% on top of the expected network delay, depending on the platform used, while still guaranteeing strong security support over the DSME network.**

I. Introduction

As the Internet of Things paradigm moves forward towards complete pervasiveness, we witness an increasing demand in terms of Quality of Service (QoS). Either for supporting scenarios in industrial automation, as key enabling technology of the industry 4.0 paradigm, or in automotive ADAS systems or even internal aircraft communications, all these scenarios pose significant stringent requirements to timeliness, energy-efficiency, and scalability, among others. However, although some of these may have been receiving significant attention in the literature, the interplay of such more traditional properties with safety and security still presents significant challenges.

In this line, the new IEEE 802.15.4 protocol, which has been considered as an important enabler for low-power and low-rate networking, incorporated the IEEE 802.15.4e amendment, which aimed at supporting increased reliability via the introduction of new MAC behaviours. One of such MAC behaviors is the Deterministic and Synchronous Multi-channel Extension (DSME) [14] which can be considered as a remake of the previous beacon-enabled mode, fitted with important reliability add-ons, aiming at minimizing interference, and maximizing the number of nodes that can be supported by its deterministic service. With these improvements, this new protocol arises as a prominent candidate to become a *de facto* standard for industrial low-power and low-rate networking systems, by focusing on improving latency, reliability and power efficiency.

However, one important drawback of this standard is the absence of efficient security schemes to ensure the confidentiality, authentication and integrity of data. In fact, certain features such as channel hopping in DSME, could be easily exploited via a quite energy efficient jamming attack, considering the channel hopping sequence is easily predictable, as is the start-time of the transmission slots. A strong security protocol in this case can secure data, but can have dire trade-off consequences in terms of the QoS.

In this paper we integrate TAKS [1] (*Topology-Authenticated Key Scheme*), a lightweight *hybrid* cryptography scheme designed specifically for WSN. TAKS keys are built from *components* which get pre-distributed according a chosen logical network topology and only the combination of the right set of components allows a WSN node to reconstruct the valid key for a given transmission.

The contribution of this paper are as follows:

- We design a network architecture for integrating a TAKS-enabled security layer over the DSME MAC layer for secure and deterministic communications, with maximum standard compatibility;
- We implement the proposed TAKS security layer over a IEEE 802.15.4e DSME MAC communication stack using the OpenDSME project [10] .
- We analyze via simulation, the significance of the introduced overhead to DSME communications over different network settings and mote platforms, in terms of application end-to-end delay and throughput.

In what follows, we provide a brief literature survey on similar enhancements provided for the IEEE 802.15.4e protocol and other security measures similar to TAKS. Then in Section III and IV we provide a detailed background to the security aspects of DSME behavior and TAKS scheme introduced in this paper. In Section V, we provide our network architecture on integrating TAKS with DSME, which is then complemented with a comprehensive performance analysis in Section VI. We wrap up this work with some discussions and future work ideas in Section VII.

II. Related works

Hybrid cryptography solutions are widely adopted in the context of low-performance embedded systems such as WSN

978-1-7281-5297-4/20/18/$31.00 ©2020 IEEE

[5] to help solve the *key distribution* and the *authentication* problems [6]. Most researches focus on providing ad-hoc solutions to optimize a pre-defined set of metrics (e.g. performance, memory occupation, energy consumption etc.) of interest. Thanks to the technology enhancement of the WSN node platforms, recent hybrid cryptography schemes started to adopt *Elliptic Curve Cryptography* (ECC) based protocols as public-key mechanism and the *Advanced Encryption Standard* (AES) as inner symmetric-key mechanism.

TAKS, the hybrid cryptography scheme we adopted in our work, has been designed and implemented to directly use the features and address the constrains of WSNs. Thanks to its low impact on memory and in performance, TAKS has been successfully adopted in the context of in European research project e.g. SafeCOP [12] [13] as security mechanism on the IEEE 802.15.4 WSNs and implemented in the context of SafeCOP Use Case 5 in in the *Road-Side Units* (RSU) components. Some of the few works aiming at security over the new IEEE 802.15.4 target LLDN based networks [11]. This work provides an insight about the proposal's impact on the Quality of Service, by analysing superframe sizes, base timeslot size and data payload, with and without security. In our work, we carry out a similar performance analysis when assessing the security overhead.

In the literature, however, there have been several key management systems [7] for the previous releases of the IEEE 802.15.4 standard. The authors use standard cryptography schemes to ensure a light-weight key management solution for these low power networks. There have been also enhancements [8] proposed to IEEE 802.15.4 to prevent same-nonce attack, denial-of-service attack, reply-protection attack and even ACK attacks. There have even been experimental validations [9] of the impact of security on a variant of standpoints such as memory consumption, network performance, and energy consumption. In this paper, we propose a security mechanism for DSME network and learn its impact on the Quality of Service of the network such as throughput and latency.

III. Background to DSME

IEEE 802.15.4 supports low rate and robust real-time communication. The Deterministic Synchronous Multi-channel Extension (DSME) of IEEE 802.15.4e stands out because of its exclusive features such as multi-channel access and *Guarantee Timeslots* (GTS) that increases the overall scalability and also provides deterministic communication. The DSME network is time-synchronized by the *Multisuperframe* structure (Figure 1). The rows that span across the Multisuperframe indicate the channels and the columns represent the timeslots. Every superframe within a Multisuperframe consists of Contention Access Period (CAP), that uses CSMA/CA for data transmission and Contention Free Period (CFP), that uses Guaranteed Timeslots (GTS). Every GTS slot accommodates the transmission of data and an eventual acknowledgment.

The superframe is defined by *BO*, the *Beacon Order* which is the transmission interval of a beacon in a superframe. *MO* is the *Multisuperframe Order* that represents the *Enhanced Beacon* (EB) interval of a multi-superframe, and *SO* is the *Superframe Order* that represents the beacon interval of a superframe. The number of superframes in a multisuperframe can be given by 2^{MO-SO}. The PAN coordinator sets the values of BO, SO, and MO. These values are conveyed to the nodes by an Enhanced Beacon at the beginning of each Multisuperframe.

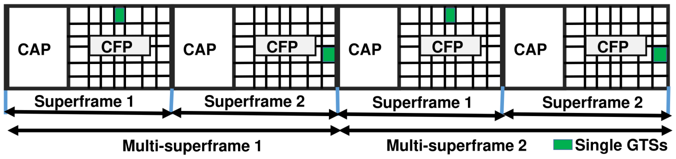


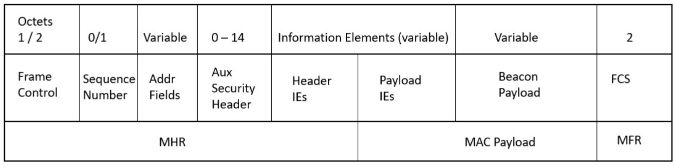
Fig. 1. IEEE 802.15.4 DSME multi superframe structure

The standard also defines a set of presets that can be much adequate for certain applications. For instance, the delay sensitive applications will need shorted superframe intervals, so that they can get accommodated immediately. Later in our performance analysis, we use similar delay sensitive settings.

Application type	BO	SO	MO	CAP reduction
Delay sensitive	6	4	6	Enabled
Reliability sensitive	8	3	4	Disabled
Energy Critical	14	1	14	Enabled
High throughput	10	5	6	Disabled
Large scale	10	1	8	Enabled

TABLE I
Standard defined presets

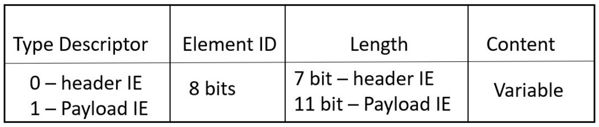
The time and slot synchronization in an IEEE 802.15.4 DSME network is maintained by an Enhanced Beacon that is issued at the start of every multisuperframe time-period. The structure of an Enhanced Beacon is presented in Figure 2. Unlike the traditional beacon used in the previous releases of IEEE 802.15.4, the EB has several capabilities such as carrying Information Elements (IE) (Figure 3). An Information Element can be a part of the MAC Header or MAC payload. A MAC header based IEs can be used by the MAC to process key functionalities like security and addressing. The MAC Payload IEs on the other hand can carry elements like routing metrics [16]. The IE can be user-defined to carry a set of information tailored for the network necessity. An IE is made up of a Type Descriptor (1 bit) to describe the type of IE, Element ID, Length, and the Content. DSME is a haven of several unique features like *CAP reduction*, *Multi-channel access*, *Group Acknowledgement* and *Fast Association* to name a few, however it lacks a credible security implementation to



Octets 1 / 2	0/1	Variable	0 – 14	Information Elements (variable)		Variable	2
Frame Control	Sequence Number	Addr Fields	Aux Security Header	Header IEs	Payload IEs	Beacon Payload	FCS
MHR					MAC Payload		MFR

Fig. 2. DSME Enhanced Beacon structure

guarantee data confidentiality and authenticity. We integrate *TAKS* with this objective and implemented it over OpenDSME. The OpenDSME [10] is a framework which provides a DSME stack implementation that can be used as a OmNet++ simulation model or deployed in real mote platforms. In OpenDSME, there are several compound modules such as the DSME data link layer which handles the GTSs allocations. One of the major limitations of OpenDSME is the lack of security modules. One of the major contribution of this work is to provide and implement a security module over OpenDSME, greatly improving the security capabilities of this project.



Type Descriptor	Element ID	Length	Content
0 – header IE 1 – Payload IE	8 bits	7 bit – header IE 11 bit – Payload IE	Variable

Fig. 3. Structure of an Information Element

IV. Background to TAKS

The *Topology-authenticated Key Scheme Version 2* (TAKS2) [1] is a *Hybrid* cryptography scheme designed for IEEE 802.15.4 Wireless Sensor Network (WSN) platforms. Figure Fig. 4 provides the block diagram of this scheme. This methodology combines the performance and reduced memory footprint of symmetric cryptography schemes (e.g. AES [4]) with a light and effective solution for key distribution based on different *key components* which are generated offline and *combined* together to produce the symmetric cryptographic key (*shared secret*, SS) used to encrypt the frame payload. Moreover, TAKS2 includes a basic *authentication* mechanism to sign and verify the transmission content. In the pair-wise version of TAKS2, the key components are two:

- The private component, pre-distributed and never transmitted (*Local Key Component*, LKC);
- The public components, pre-distributed and available to every node logically enabled to communicate with target node (*Transmit Key Component*, TKC).

In order to generate and to combine key components, TAKS2 exploits vector algebra on top of a prime or a finite field (e.g. $GF(2^n)$). The combination of key components is done by an internal function called $TAK()$ which takes, as input, the LKC of the sending node, the TKC of the destination node and a *nonce* value. The reference $TAK()$ function is the following:

$$SS_{i \to j} = \alpha LKC_i \times TKC_{i \to j} = -\alpha TKC_{i \to j} \times LKC_j$$

The functions to perform (symmetric) encryption and *Message Authentication Code* computation, the key components length (in bits) and the resulting symmetric key can be arbitrarily chosen, although key sizes compatible with the encryption function are preferred.

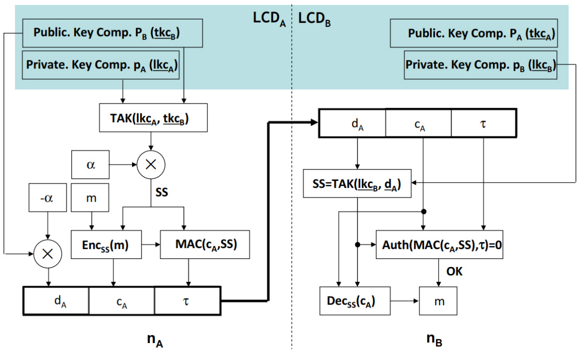


Fig. 4. TAKS2: Block diagram for sender (left) and receiver (right)

V. Integration of TAKS with DSME

A. TAKS Implementation

The TAKS Key Components are represented by the generic class `TaksComponent<>`. This class is responsible of the storage, the retrieval and the serialization of the data of a single key component (e.g. LKC or TKC). In order to reduce the memory required to store components and to speed up the generation of the shared-secret used as symmetric key, the vector dimension of the TAKS Key Components has been selected to be 2, which is the minimum dimension compatible with the TAKS computations i.e. each TaksComponent is a vector (x, y) in which each components size depends on the selected key length, e.g. 128 bit.

The high-level encryption and the decryption functions (as in Fig. 4) are implemented in the singleton class `Taks`. This class:

1) generate the nonce from a Cryptographically-Secure pseudo Random number Generator (CSRNG);
2) provide the implementation of the TAK function, the $GF(2^8)$ arithmetics and the vector product of `TaksComponents`;
3) delegate the symmetric encryption (with SS as key) to an external symmetric cipher.
4) delegate the MAC calculation (with SS as key) to an external MAC function.

For sake of example, we have provided this implementation with a simple XOR-based symmetric cipher (as symmetric cipher) and a simple checksum-based function (as MAC function). However, as the IEEE 802.15.4 standard suggests ([3] Annex B), it is possible to adopt AES [4] with a key-length of 128 bits in CCM* mode (which provides both encryption/decryption and an authentication tag). Finally, we provided a separated *configuration* unit to allow users to enable/disable security and select different size for key components, symmetric keys and MAC.

B. IEEE 802.15.4 Headers

As Fig. 4 shows, in addition to the encrypted message content (the ciphertext, c_A) two other fields have to be added to allow the receiver to authenticate and decrypt the frame. Following the IEEE 802.15.4 standard, we decided to:

1) implement the optional *Auxiliary Security Header*;
2) investigate the *Information Element* (IE) structures;
3) append the MAC (τ) and the key reconstruction information (d_A) as IE.

1) Auxiliary Security Header: This header is optionally defined in the standard MAC-layer header of a IEEE 802.15.4-conformant frame. The presence of this header is signaled by the `SecurityEnabled` bit inside the *FrameControl* field of the MAC-layer header. If the `SecurityEnabled` bit is 1, the receiver should expect the presence of the *Auxiliary Security Header* after the `SourceAddress` field (Section 7.2 of [3]). The *Auxiliary Security Header* contains three fields:

1) the (mandatory) `SecurityControl` field, which describes the security level adopted (e.g. key length) and signals the presence of the optional fields given below;
2) the (optional) `FrameCounter` field, which is used both as (part of the) nonce in the CCM* mode [1];
3) the (optional) `KeyIdentifier` field, which contains indications on where the key should be retrieved.

In our implementation, since the nonce and the adopted symmetric key is generated by TAKS, we construct a *Auxiliary Security Header* with a `SecurityControl` field with no additional fields. We added the missing `AuxiliaryHeader` class to openDSME to take care of the storage and serialization of the *Auxiliary Security Header*.

2) TAKS Information Element: The format of IEEE 802.15.4 IEs is complex and it is out of the scope of this paper. However, to motivate the implementation decisions, we briefly report some requirements for having IEs in frame:

- The presence of IEs is signaled in *FrameControl* field of the MAC-layer header;
- IEs are optional and can be of two types: *Header IE*s and *Payload IE*s;
- Depending on the IE type, two different structures has to be adopted and, more important, proper *Header Terminations* fields (*HT1*, *HT2* and the *Payload Termination*) should be used so that the receiver can decode the frame fields correctly.

We analyzed the different possibilities for storing the MAC (τ) and the key reconstruction information (d_A) as IE and, focusing on reducing the bytes required to store such fields, we decided to implement both d_A and τ as *Header IE*s with custom *Element ID*, since TAKS has not yet proposed to the IEEE registration authority. In order to provide openDSME with the concept of IE with all the standard features, we designed a class hierarchy consisting of an common interface (`IE`), two abstract implementation of it (`HeaderIE` and `PayloadIE`) and, finally, a concrete class deriving from `HeaderIE` called `TAKS_IE` to manage and provide serialization of the two IEs (one for d_A and another for τ).

C. Payload Representation

The representation of a generic payload of a frame (which is not provided by openDSME) has been implemented by a new class (`GenericPayload`) which takes care of storing and retrieving a user-defined payload in a frame.

D. Integration in openDSME

The overall integration of TAKS and openDSME is shown in Fig. 6. In order to integrate TAKS in openDSME, we have modified the openDSME code responsible of the management of outgoing and incoming frames to include (when enabled) the TAKS functionalities. In particular:

- when a outgoing frame arrives to the MAC-layer, our `TAKS` class is invoked to:
  1) creating a `GenericPayload` from the frame payload;
  2) construct a valid *Auxiliary Security Header*;
  3) retrieve the right set of key components depending on the ID of the sender/receiver;
  4) perform the *TAKS Encryption* (Fig. 4) on the payload;
  5) computing the d_A and the τ and to create their IEs;
  6) serialize everything to obtain a valid IEEE 802.15.4 frame
- upon reception of a frame, our `TAKS` class is invoked again to:
  1) retrieve the LKC (required for decryption);
  2) parse the IEs and the payload, obtaining the d_A and τ sent;
  3) authenticate the frame by computing the MAC and comparing it with τ). If the check fails, the frame is dropped;
  4) using d_A, compute back the SS and perform the *TAKS Decryption* (Fig. 4) on the payload;

The overall structure of the frames is shown in Fig. 5.

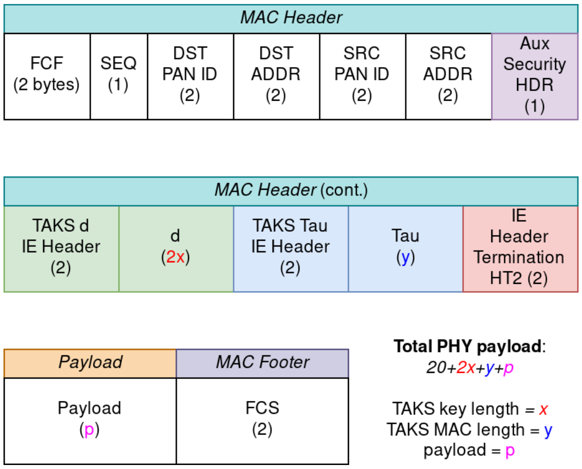


Fig. 5. Frame structure adopted in our implementation

VI. Performance analysis

Although TAKS provides a significant step forward in terms of security for DSME networks, one must careful analyse its overhead to pinpoint or bound any inherent limitation to its

[1] The standard assumes that the encryption to be used is AES in CCM* mode, with a maximum key length of 128 bits

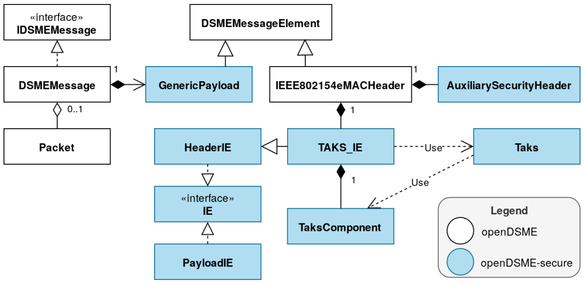


Fig. 6. Implementation: OpenDSME components (in white) and the components added in *OpenDSME-secure*

application. In this section, we will focus on evaluating this and importantly, the impact or significance of such overhead upon different network settings. We relied on the OpenDSME stack implementation, deployed in the OmNet++ network simulator. Let us consider any TAKS-enabled DSME network, in which traffic is directed at a sink, i.e. let us assume the PAN Coordinator takes such role, the total payload data P carried in a DSME frame can be given by:

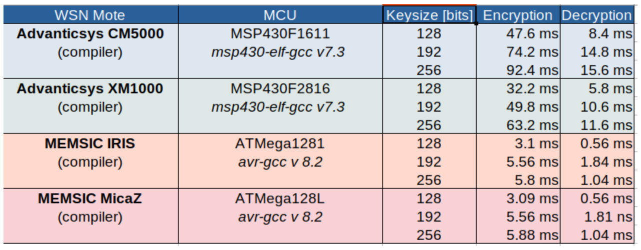
$$P := \sum_{i=0}^{card(D)} payload(d_i) \tag{1}$$

where, d_i is a data frame which is a part of the total number of frames that is transmitted. Considering $txrx(d_i)$ is the end to end transmission for a data frame and $t_{enc,i}$ and $t_{dec,i}$ are their respective encryption and decryption times, the total time spent in the transmission of the data frame is expressed as:

$$T_D := \sum_{i=0}^{card(D)} t_{enc,i} + txrx(d_i) + t_{dec,i} \tag{2}$$

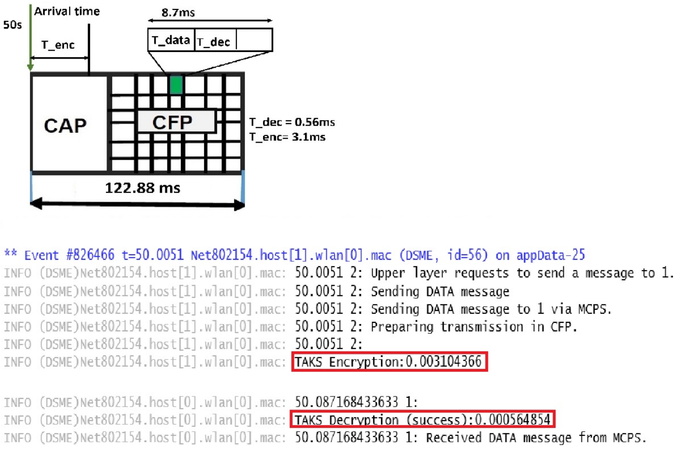
Assuming all DSME traffic is transmitted using its GTS service, $txrx(d_i)$ delay bound is deterministic. However, the delay introduced by encryption and decryption operations, is dependent on the WSN-platform used. We implemented the TAKS encryption/decryption procedures in different popular WSN-platforms and computed their duration for fixed packet lengths. As an example, we provide in Figure 7, a table of the encryption and decryption delay at different key lengths. As expected, delays are quite dependant on the platform, being exceptionally worse for the Advanticsys platforms, and grow according to the key length used. The computed values on these platforms were introduced into the OmNet++ implementation, to re-create these differences in the simulation analysis of these networks, addressing the $t_{enc,i}$ and $t_{dec,i}$ mentioned above. To complement these observations within the networking context, and ease the comprehension of these networking properties, we depict in Figure 8 how these delays come into play together.

We assume a DSME network with a setting of MO=3, SO=3 and BO=6, so that one superframe repeats every multisuperframe interval with a duration of 122.88ms [15]. We provide a representation of the superframe, with the corresponding transmission GTS slot marked in green. Alongside, we represent the packet arrival time, and additional delays. To complement this information, we provide a screen capture from the simulator, in which we highlight the timestamps of the packet encryption, reception, and finally of its decryption and arrival at the application layer. In the forthcoming experiments, we shall look in detail into these delays, and consider their significant at different network setting.



WSN Mote	MCU	Keysize [bits]	Encryption	Decryption
Advanticsys CM5000 (compiler)	MSP430F1611 *msp430-elf-gcc v7.3*	128	47.6 ms	8.4 ms
		192	74.2 ms	14.8 ms
		256	92.4 ms	15.6 ms
Advanticsys XM1000 (compiler)	MSP430F2816 *msp430-elf-gcc v7.3*	128	32.2 ms	5.8 ms
		192	49.8 ms	10.6 ms
		256	63.2 ms	11.6 ms
MEMSIC IRIS (compiler)	ATMega1281 *avr-gcc v 8.2*	128	3.1 ms	0.56 ms
		192	5.56 ms	1.84 ms
		256	5.8 ms	1.04 ms
MEMSIC MicaZ (compiler)	ATMega128L *avr-gcc v 8.2*	128	3.09 ms	0.56 ms
		192	5.56 ms	1.81 ns
		256	5.88 ms	1.04 ms

Fig. 7. Encryption and Decryption times of WSN compilers



```
** Event #826466 t=50.0051 Net802154.host[1].wlan[0].mac (DSME, id=56) on appData-25
INFO (DSME)Net802154.host[1].wlan[0].mac: 50.0051 2: Upper layer requests to send a message to 1.
INFO (DSME)Net802154.host[1].wlan[0].mac: 50.0051 2: Sending DATA message
INFO (DSME)Net802154.host[1].wlan[0].mac: 50.0051 2: Sending DATA message to 1 via MCPS.
INFO (DSME)Net802154.host[1].wlan[0].mac: 50.0051 2: Preparing transmission in CFP.
INFO (DSME)Net802154.host[1].wlan[0].mac: 50.0051 2:
INFO (DSME)Net802154.host[1].wlan[0].mac: TAKS Encryption:0.003104366

INFO (DSME)Net802154.host[0].wlan[0].mac: 50.087168433633 1:
INFO (DSME)Net802154.host[0].wlan[0].mac: TAKS Decryption (success):0.000564854
INFO (DSME)Net802154.host[0].wlan[0].mac: 50.087168433633 1: Received DATA message from MCPS.
```

Fig. 8. superframe structure for a 633 setting with encryption and decryption times for the MEMSIC IRIS mote for a 128 bit security key

A. Packet Length impact due to security overhead

As illustrated in Figure 5, there is a clear security overhead on the exchanged frame, which increases with the key length. As we will analyse, this overhead impacts the maximum amount of application-relative data that can be exchanged, particularly considering that a GTS slot, has a well-defined and limited space consisting of a minimum of one timeslot duration. Considering such frequently used allocation aiming at maximizing the accommodation of nodes into a single superframe, we are unable to use a *SO* setting since there is not enough space in such slot to accommodate the packet lengths enforced by TAKS implementation. Table II shows the maximum application data payload we can fit into a GTS slot, for different SO settings and security key lengths.

As we can observe, a SO value of 2 can't fit more than 22 bits of application data in one GTS slot, even considering no TAKS security overhead exists. Thus, it is not possible

	No security	64 bits	128 bits	192 bits	256 bits
622	22B	X	X	X	X
633	75B	75B	75B	63B	63B
644	75B	75B	75B	75B	75B

TABLE II
Maximum Application packet size vs Security key lengths

to rely on TAKS to send any application payload for $SO < 3$. Due to this reason, for $SO = 3$, the user is limited to a security key length of 128 bits if aiming to send 75 Bytes of application data. However, to secure our data with a 192 or 256 key length, we are unable to go beyond 63 Bytes of application data for this specific SO setting. For higher SO values i.e. $SO > 4$, this does not represent a limitation any further, and the user is able to use the maximum application payload size, while using the largest security key lengths. However, this limits the application end-to-end delay, as the user is prevented from relying upon the network settings that could provide the minimum delay.

B. Application end-to-end delay analysis

Naturally, application end-to-end delay is a metric that heavily defines the limits of an application, particularly for critical industrial scenarios, and thus must be carefully analyzed in regards to the influence of both DSME settings and security encryption and decryption procedures. We consider a DSME network composed of five nodes, that rely on five GTS slots allocated for periodic transmissions e.g. every superframe at a fixed traffic period of 0.5s. We simulated this network for different settings i.e. $\langle BOSOMO \rangle = 633, 644, 655$, and for scenarios without TAKS, and with TAKS using different key lengths. Figure 9 presents the maximum application end-to-end delays. As expected, due to the increasing superframe sizes, network delay increases for higher SO settings. Depicted on top of the network delay, in black, we see the delay relative to the encryption and decryption procedure overhead. Interestingly, for $SO = 3$, this delay is not negligible, and is quite close to the delay introduced by the network. The contribution of the delay increases with the length of the security key used, however its impact in the overall delay is less significant as the SO increases. The presented application end-to-end delays are naturally dependant on the WSN platform and previously shown. In Figure 10 we analyze the application end-to-end maximum delays for one of these network scenarios i.e. $\langle 633 \rangle$, considering different mote platforms. As observed, the impact of the delay introduced by the security overhead is much more significant for the Advanticsys platforms, and for such settings is considerably higher than the delay introduced by the network. This issue must be considered when carrying out a rigorous application planning, and this simulation model offers an important contribution in that line, by implementing a security approach along with the possibility to assess its overhead on real WSN platforms before deployment.

C. Security impact on network throughput

As with application end-to-end delay, the security implementation overhead is expected to have an impact on the network throughput as well. Due to the encryption overhead, generated packets will wait for a considerable higher amount of time before being en-queued for transmission, which will reflect in the amount of traffic delivered into the network.

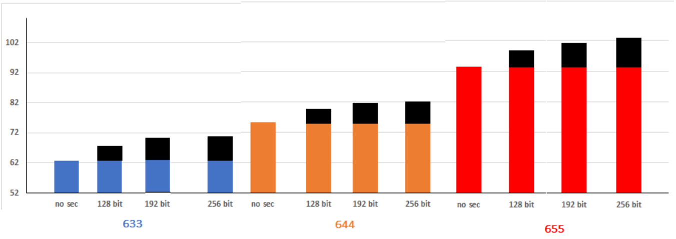


Fig. 9. Application maximum end-to-end delay in ms for different settings and respective security impact

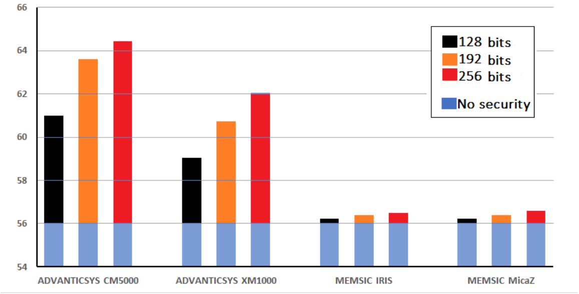


Fig. 10. Application maximum end-to-end delay in ms for different motes and variable security key lengths using $SO = 3$

For the following experiment we setup a scenario of BO=6, MO=3 and SO=3, with 2 nodes, each with 3 GTS Slot allocated, transmitting at a fixed rate of 25Hz (40 ms Period). As depicted in Figure 11, as expected from the previous observations, the chosen platforms, along with the security key lengths clearly affect this metric. The higher the encryption delays introduced by TAKS, the lower the network throughput will be, as packets must wait for the encryption procedure to complete to be en-queued for transmission. Interestingly, we observe that if encryption delays are equal or lesser than the application data generation period, this results in barely no impact on throughput. This is more visible in Figure 12 in which we present the application throughput results as we increase the traffic generation rate, for two mote platforms and using different key lengths. As shown, increasing traffic generation rates generate increasing application throughput, until a saturation occurs, due to the limit amount of traffic the GTS service can avail. Saturation occurs earlier for those specific scenarios in which encryption delay is higher. As for higher SO settings, no change in throughput was observed.

VII. Conclusion and Future Work

In this paper, we introduced TAKS, a hybrid cryptography scheme which is an effective scheme for key distribution, providing security and reduced memory footprint. We integrated it onto the security layer of the IEEE 802.15.4 DSME MAC behaviour and effectively implemented it over a network

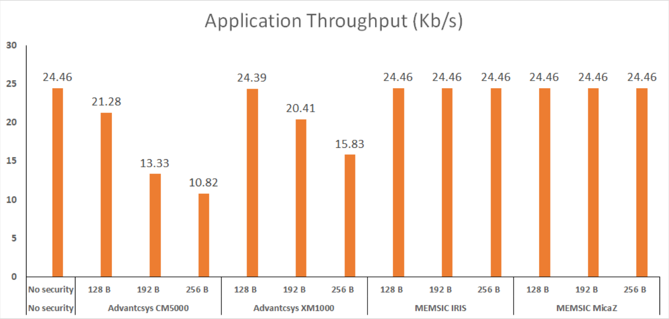


Fig. 11. Network throughput comparison for security implementations in different motes

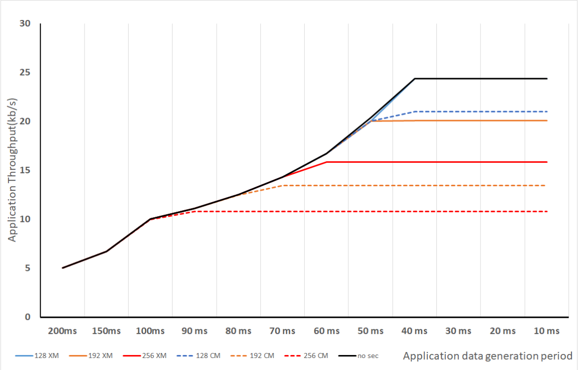


Fig. 12. Network throughput analysis according to different Data rates and variable security key lengths for Advanticsys motes

simulator. We relied on this simulation model and on partial real mote implementation results to evaluate its overhead for critical DSME network settings and predict its performance on different WSN platforms. Finally, we presented the impact and relative significance of its overhead, showing that low application delays in the order of 56 ms can still be achieved. However, particular attention must be given to the overhead introduced by each particular platform, as this alone can introduce an additional delay in the order of 1% to 14,3%. In addition, the effect in throughput is limited to very specific scenarios of low SO where the application data generation period is smaller than the time it takes the WSN node to process and complete the encryption of the packet. We believe this is effort constitutes a significant step towards the security tightening of IEEE 802.15.4 networks, particularly to those relying on the flexibility of the DSME MAC behaviour. The implemented simulation model can support an effective evaluation of the network behaviour we can expect using different mote platforms, which is mandatory for more critical scenarios, considering the encryption and decryption procedures overhead. We expect to extend and analyse this solution in multi-channel network scenarios, taking additional advantages from such DSME functionalities. We also plan to implement the full protocol stack and security solution in off the shelf platforms, pushing forward the usage of such technologies on different critical industrial scenarios.

Acknowledgment

This work was partially supported by National Funds through FCT/MCTES (Portuguese Foundation for Science and Technology), within the CISTER Research Unit (UIDB/04234/2020).

References

[1] M. Pugliese and F. Santucci, ”Pair-wise network topology authenticated hybrid cryptographic keys for Wireless Sensor Networks using vector algebra,” 2008 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems, Atlanta, GA, 2008, pp. 853-859. doi: 10.1109/MAHSS.2008.4660137

[2] L. Pomante, M. Pugliese, S. Marchesani and F. Santucci, ”WINSOME: A middleware platform for the provision of secure monitoring services over Wireless Sensor Networks,” 2013 9th International Wireless Communications and Mobile Computing Conference (IWCMC), Sardinia, 2013, pp. 706-711. doi: 10.1109/IWCMC.2013.6583643

[3] IEEE Standard for Low-Rate Wireless Networks,” in IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011) , vol., no., pp.1-709, 22 April 2016 doi: 10.1109/IEEESTD.2016.7460875

[4] Advanced Encryption Standard (AES), Federal Inf. Process. Stds. (NIST FIPS) - 197, https://doi.org/10.6028/NIST.FIPS.197

[5] R. B. Gandara, G. Wang and D. N. Utama, ”Hybrid Cryptography on Wireless Sensor Network: A Systematic Literature Review,” 2018 International Conference on Information Management and Technology (ICIMTech), Jakarta, 2018, pp. 241-245. doi: 10.1109/ICIMTech.2018.8528147

[6] Simplicio, Marcos and Barreto, Paulo and Margi, Cintia and Carvalho, Tereza. (2010). A survey on key management mechanisms for distributed Wireless Sensor Networks. Computer Networks. 54. 2591-2612. 10.1016/j.comnet.2010.04.010.

[7] IRaza, Shahid, Thiemo Voigt, and Vilhelm Jutvik. ”Lightweight IKEv2: a key management solution for both the compressed IPsec and the IEEE 802.15. 4 security.” Proceedings of the IETF workshop on smart object security. Vol. 23. 2012.

[8] Xiao, Yang, et al. ”Security services and enhancements in the IEEE 802.15. 4 wireless sensor networks.” GLOBECOM’05. IEEE Global Telecommunications Conference, 2005.. Vol. 3. IEEE, 2005.

[9] Daidone, Roberta, Gianluca Dini, and Marco Tiloca. ”On experimentally evaluating the impact of security on IEEE 802.15. 4 networks.” 2011 International Conference on Distributed Computing in Sensor Systems and Workshops (DCOSS). IEEE, 2011.

[10] Kstler, Maximilian, Florian Kauer, Tobias Lbkert, Volker Turau, J. Scholz, and A. von Bodisco. ”Towards an open source implementation of the IEEE 802.15. 4 DSME link layer.” Proceedings of the 15. GI/ITG KuVS Fachgesprch Sensornetze, J. Scholz and A. von Bodisco, Eds. University of Applied Sciences Augsburg, Dept. of Computer Science (2016).

[11] Anwar, Mashood. et al ”TDMA-based IEEE 802.15. 4 for low-latency deterministic control applications.” IEEE Transactions on Industrial Informatics 12.1 (2015): 338-347.

[12] SafeCOP project website, URL: http://http://www.safecop.eu

[13] SafeCOP Report on T3.2 on safe and secure communication protocols for the different use cases and demonstrators, URL: http://www.safecop.eu/wp-content/uploads/2019/01/Report-on-T3.2-on-safe-and-secure-communication-protocols-for-the-different-use-cases-and-demonstrators.pdf

[14] Kurunathan, Harrison, et al. ”IEEE 802.15. 4e in a nutshell: Survey and performance evaluation.” IEEE Communications Surveys Tutorials 20.3 (2018): 1989-2010.

[15] Kurunathan, Harrison, et al. ”Worst-case bound analysis for the time-critical MAC behaviors of IEEE 802.15. 4e.” 2017 IEEE 13th International Workshop on Factory Communication Systems (WFCS). IEEE, 2017.

[16] Kurunathan, Harrison, et al. ”Symphony: routing aware scheduling for DSME networks” ACM Sigbed Review, ACM. Dec 2019, Volume 16, Issue 4, pp 26-31. Special Issue on International Workshop on Real-Time Networks (RTN 19).