Application-layer security for the WoT: Extending CoAP to support end-to-end message security for Internet-integrated sensing applications

Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva

DEI/CISUC, University of Coimbra
Polo 2, Pinhal de Marrocos, 3030-290 Coimbra, Portugal
{jgranjal, edmundo, sasilva}@dei.uc.pt

Abstract. Future Web of Things (WoT) applications employing constrained wireless sensing devices will require end-to-end communications with more powerful devices as Internet hosts. While the Constrained Application Protocol (CoAP) is currently being designed with this purpose, its current approach to security is to adopt a transport-layer solution. Transport-layer security may be limitative, considering that it does not provide a granular and flexible approach to security that many applications may require or benefit from. In this context, we target the design and experimental evaluation of alternative security mechanisms to enable the usage of end-to-end secure communications at the application-layer using CoAP. Rather than replacing security at the transport-layer, it is our goal that the proposed mechanisms may be employed in the context of a broader security architecture supporting Internet-integrated wireless sensing applications. Ours is, as far as we known, the first proposal with such goals.

Keywords: CoAP security, DTLS, end-to-end application-layer security, message security, granular security.

1 Introduction

Although many of the applications currently envisioned for the Web of Things (WoT) are critical in respect to security, the fact that they are envisioned to employ very constrained sensing platforms and wireless communications complicates the design of appropriate security solutions. In practice, many applications are required to accept compromises between security and the usage of resources on constrained sensing platforms. With wireless sensing devices as the TelosB [1] energy is a scarce resource. Such devices are employed in the context of low-energy personal area networks (LoWPANs) using link-layer communications standards such as IEEE 802.15.4 [2]. IEEE 802.15.4 supports low-energy wireless communications at low transmission rates using small packets in order to minimize transmission errors, and technologies for the integration of LoWPANs with the Internet are starting to appear and are expected to play an important role in the fulfillment of the WoT vision.

Communications and security technologies for the WoT are currently in the design phase and consequently a communications and security architecture for the WoT is currently not completely defined. In this context, of particular relevance are technologies currently being designed at the 6LoWPAN (IPv6 over Low Power Personal Area Networks) [3] and Core (Constrained RESTful Environments) [4] working groups of the IETF. Such technologies target the usage of LoWPAN devices in the context of its integration with the Internet, and as such are of most relevance to the WoT. 6LoWPAN provides adaptation mechanisms to enable the transmission of IPv6 packets over LoWPAN environments as IEEE 802.15.4 [2], while CoRE is currently designing the Constrained Application Protocol (CoAP) [5] with the purpose of enabling RESTful HTTP-based web communications on such environments. Focusing on how CoAP approaches security, we observe that the current choice to support end-to-end security is to adopt the Datagram Transport Layer Security (DTLS) Protocol [6]. Thus, security is not integrated at the application-layer protocol itself, but rather transparently applied to all CoAP messages at the transport layer. Given that 6LoWPAN environments currently employ UDP, DTLS appears as a logical choice in protecting communications at higher layers, at least from the standpoint of standardization. As we address in the paper, this approach misses all the advantages available in the usage of security at the application layer. With this in mind, we propose and experimentally evaluate new security mechanisms for the CoAP application-layer protocol.

The paper proceeds as follows. Section II describes related work, and Section III discusses end-to-end security in the context of Internet-integrated LoWPANs using 6LoWPAN and CoAP. The proposed mechanisms are described in Section IV and experimentally evaluated in Section V. Finally, Section VI concludes the paper.

2 Related Work

Although security for Wireless Sensor Networks (WSN) is a prolific research area, investigation concerning the integration of LoWPAN environments with the Internet is very recent, and in consequence less proposals are available that target security in this context, and in particular security for end-to-end communications between LoWPAN wireless sensing devices and Internet hosts. Nevertheless, we find it important to address proposals with goals similar as ours, even if not addressing application-layer security using the (currently being designed) CoAP protocol.

The first of such proposals is Sizzle [7], which implements a compact web server providing HTTP accesses protected by SSL using 160-bit ECC (Elliptic Curve Cryptography) keys for authentication and key negotiation, but requiring a reliable transport-layer protocol and therefore being incompatible with CoAP and 6LoWPAN. Sizzle also does not support two-way authentication as will be required by many Machine-to-Machine (M2M) applications on the WoT. On the other end, SSNAIL [8] supports two-way authentication using an ECC-enabled handshake, but also requiring a reliable transport-layer protocol. More in line with the 6LoWPAN and CoAP technologies, authors in [9] propose the compression of DTLS headers with the goal of saving payload space and in consequence reducing the communications overhead. The architecture proposed in [10] supports two-way authentication with DTLS for end-to-end communications with constrained sensing devices employing specialized trustedplatform modules (TPM) to support hardware-assisted RSA cryptography. A previous internet-draft [11] proposed the integration of security with CoAP using options for the activation/deactivation of security contexts and for the protection of CoAP messages. Although this proposal shares some goals with ours, it assumes that all exchanged CoAP messages are protected in a similar fashion, as security is handled in the context of security sessions previously established between CoAP communicating entities. We may thus consider that this proposal is more in line with transport-layer security than with how application-layer security may be handled for individual CoAP messages. Other limitation of this proposal is that it doesn't enable a CoAP message to securely transverse multiple trust domains. Overall, none of the previously discussed proposals addresses application-layer security using CoAP with our goals.

3 End-to-end communications for Internet-integrated wireless sensing and actuating applications

The current Internet architecture illustrates the importance of employing complementary security mechanisms at diverse protocol layers. This aspect may also be considered when planning security for Internet-integrated LoWPANs, and in particular regarding the protection of end-to-end communications. End-to-end transport-layer security using DTLS as currently proposed for CoAP may be appropriate to applications requiring the transparent encryption of all CoAP communications, while on the other side applications may benefit from a more granular approach to security. Applications may require that security be applied according to the semantics of the CoAP protocol, or to the type of message or its contents. Overall, different approaches to end-to-end security may not only enrich the set of solutions available for Internet communications in the context of Internet-integrated LoWPANs, but also contribute to a more intelligent allocation of resources to security, given the computational and energetic impact of the cryptographic operations. Before proceeding with a discussion on how we approach application-layer CoAP security, we find it important to discuss how 6LoWPAN and CoAP are employed to support end-to-end communications with external (Internet) devices, as Figure 1 illustrates.

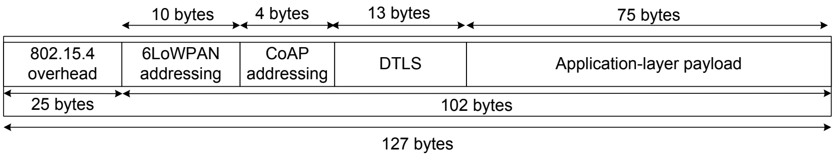

Fig. 1. Payload space usage for end-to-end communications in IEEE 802.15.4 environments.

We may observe that payload space is a scarce resource in LoWPAN IEEE 802.15.4 environments, and as a consequence 6LoWPAN and CoAP incorporate header and address compression whenever viable. IEEE 802.15.4 provides 127-bytes of payload space at the link-layer, from which 25 bytes are required for the purpose of link-layer addressing. Therefore, 102-bytes of payload space are available for the 6LoWPAN adaptation layer and Protocols such as DTLS and CoAP at above layers. 6LoWPAN IPHC shared-context header compression [12] enables the compression of the

UDP/IPv6 header down to 10 bytes, while CoAP employs a 4-byte fixed header and DTLS a 13-byte header. Without transport-layer security 88 bytes are available for applications using CoAP without incurring in costly 6LoWPAN fragmentations.

3.1 The CoAP Protocol

The CoAP Protocol [5] provides a request/response communications model between application endpoints and enables key concepts of the web such as the usage of URIs to identify resources in LoWPAN wireless sensing devices. In the context of an Internet-integrated LoWPAN sensing application, end-to-end communications may take place purely with CoAP or in alternative by translating HTTP to CoAP at a reverse or forward proxy, for example supported by a 6LBR (6LoWPAN border router). Such proxy entities as employed by CoAP may also be used in the benefit of security, as we discuss next in the context of our proposal. A CoAP request requiring an acknowledgment may be sent in a confirmable message, while data for which eventual delivery is sufficient may be sent in a non-confirmable message. A reset message may also be sent to identify unavailable resources or error conditions. Similarly to HTTP, CoAP also defines a set of method and response codes.

An important concept of CoAP is that, other than a basic set of information, most of the information is transported by options. CoAP options may be critical, elective, safe or unsafe. In short, a critical option is one that an endpoint must understand, while an elective option may be ignored by an endpoint not recognizing it. Safe and unsafe options determine how an option may be processed by an intermediary entity. An unsafe option needs to be understood by the proxy in order to safely forward it, while a safe option may be forwarded even if the proxy is unable to process it.

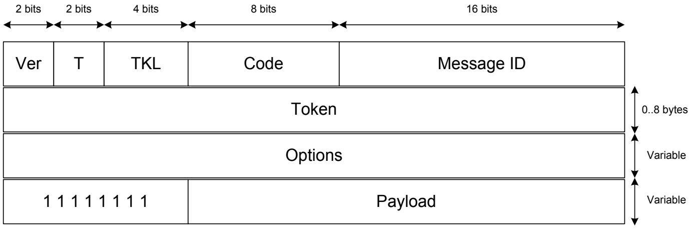

Fig. 2. CoAP message.

The CoAP header and message format as currently proposed [5] is illustrated in Figure 2. The top row illustrates the 4-byte CoAP fixed header, constituted by the version field (2 bits), the message type field (2 bits), the token length field (4 bits), the Code field (8 bits) and the Message ID (16 bits). The token enables a CoAP entity to perform request/reply matching, while the message ID field may enable duplicate and optional reliability. Each option instance in a CoAP message specifies the Option Number of the CoAP option, the length of the Option Value and the Option Value itself. CoAP options are employed to support mechanisms designed at the application-layer, and new options can be introduced to support new functionalities in the future.

3.2 Limitations of CoAP transport-layer security

The current CoAP specification [5] defines bindings to the DTLS (Datagram Transport-Layer Security) Protocol in order to enable security at the transport-layer. DTLS may apply security to all messages in a given security session, thus providing confidentiality, authentication and integrity for all CoAP communications. While DTLS is a good choice in respect to its support of efficient AES/CCM cryptography as available at the hardware in IEEE 802.15.4 sensing platforms, we may identify a few aspects motivating our alternative approach:

- Security is transparently applied to all CoAP messages: DTLS security is applied to all messages of a given communication session. A cipher suite is negotiated during the DTLS handshake and is employed to protect all CoAP messages, irrespective of the semantics of the Protocol or the type and contents of the messages. Applications are thus unable to define granular security policies and security may be more costly than what would be required by applications.

- Applications are required to employ a static security configuration: After the DTLS handshake all messages are protected using a particular cipher suite and the corresponding cryptographic algorithms and keys. Applications are thus unable to employ different security algorithms and keys to protect different messages in the context of a single CoAP wireless sensing application.

- Security is incompatible with the employment of CoAP intermediaries: Although CoAP defines the usage of proxies in forward and reverse modes [5], end-to-end security as currently proposed at the transport-layer is problematic in this context. Although end-to-end communications are at the hearth of IPv6, the exposure of constrained LoWPANs to the Internet is likely to require appropriate protection mechanisms based on the usage of security gateways. Such gateways may also support the 6LBR and CoAP proxy roles, thus breaking DTLS security. Other aspect is that such gateways may provide a strategic place for the support of heavy cryptographic operations offloaded from constrained sensing devices.

We believe that application-layer message security may address the previous discussed limitations of transport-layer security. Rather than constituting a panacea, application-layer CoAP security may complement DTLS in supporting effective end-to-end secure communications for Internet-integrated LoWPANs, according to the requirements of particular wireless sensing application.

4 CoAP application-layer message security

Our proposed mechanisms to integrate security at the application-layer with the CoAP Protocol target the issues previously discussed and may provide various benefits, which we also address in the context of the experimental evaluation of our proposal. Packet payload space usage is one aspect to address, as security-related information at the application-layer may be transported in the same context as headers and control information of the CoAP protocol itself. The overhead in terms of the required energy and computational time on constrained sensing devices is also worth investigating, given the significance of such aspects on the lifetime and the communications rate of wireless sensing applications. We proceed by describing the format and usage of the new CoAP security options. All such options are critical, unsafe and non-mandatory, given that applications may opt for security mechanisms at different layers (DTLS at the transport-layer or IPSec at the 6LoWPAN adaptation layer, for example).

4.1 *SecurityOn* CoAP security option

The *SecurityOn* option states that the given CoAP message is protected by application-layer security. The format of this option is illustrated in Figure 3. This option states the following about a CoAP message: how security is applied, what entity should process or verify security for the message, the security context that the message belongs to and temporal information relevant to ascertain about the validity of the message. CoAP options are formatted in the TLV (Type, Length, Value) format and thus the length of the *Destination Entity* field in Figure 3 may be obtained from the total length of the option.

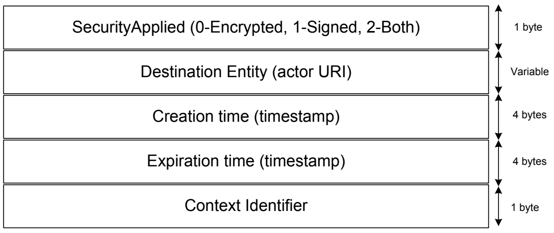
SecurityApplied (0-Encrypted, 1-Signed, 2-Both)
1 byte
Destination Entity (actor URI)
Variable
Creation time (timestamp)
4 bytes
Expiration time (timestamp)
4 bytes
Context Identifier
1 byte

Fig. 3. *SecurityOn* CoAP security option.

The *Destination Entity* field identifies the actor CoAP URI (in the form of a NULL-terminated string) that the destination must handle. This option enables the usage of application-layer security in scenarios where security associations may or may not be handled in an end-to-end fashion. The actor URI may identify the final entity receiving the CoAP message or on the other end an intermediary, thus enabling the usage of CoAP secure communications that are managed by an intermediary. This field thus states "this CoAP secured message is meant for any endpoint acting in the capacity indicated by this URI". This option may be employed more than once in a given CoAP message to enable the transversal of different trust domains possibly using also different encryption keys. The *SecurityOn* option also transports temporal values that enable verifying the legitimacy of the message. The creation and expiration time of the message are inserted by its creator and may enable an intermediary or the final CoAP ascertain the validity of the message. The context identifier enables the client, server and/or intermediaries to contextualize the message in terms of security, in particular in determining the appropriate ciphers and keys.

4.2 *SecurityToken* CoAP security option

The *SecurityToken* option enables the usage of identity and authorization mechanisms at the application-layer, on a per message basis. Using this option a CoAP requestor (client) may state "who am I" and "what I know" in order to obtain access to a given CoAP resource. With granular security applications may provide accesses to CoAP resources with different criteria, according to the identity of the client and to the criticality of the sensing data requested. Thus, although a security context between communicating entities is required, this option enables request authorization on a per message basis, thus contributing to the implementation of more detailed security policies. The format of this option is illustrated in Figure 4.

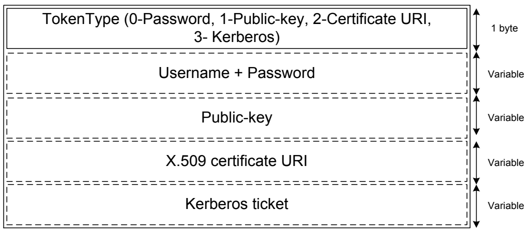

Fig. 4. *SecurityToken* CoAP security option.

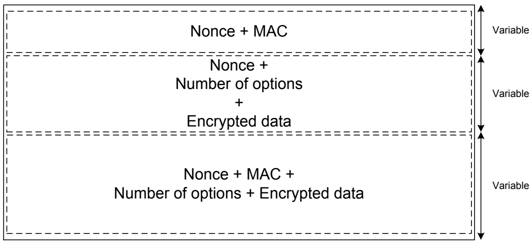
Nonce + MAC
Variable
Nonce + Number of options + Encrypted data
Variable
Nonce + MAC + Number of options + Encrypted data
Variable

Fig. 5. *SecurityEncap* CoAP security option.

A CoAP message only transports data related with one particular authorization mechanism at a time, and thus the length of the corresponding field is obtained from the total length of the option. A CoAP destination or intermediary entity along the path of the message may enforce the usage of a *SecurityToken* option in order to authorize CoAP requests. As Figure 4 illustrates, the currently defined format for this options enables a client to authenticate itself using a simple username plus password scheme, using its public-key, its X.509 certificate referred by a URI (NULL-terminated string), or a *kerberos* ticket previously obtained form a domain server (in binary format). Further authorization mechanisms may be designed or adopted in the future by defining appropriate identification values and the format of the authorization data to be transported.

A CoAP requestor may be authorized at a destination or intermediary using its public-key or X.509 certificate to validate an encrypted MAC (Message Authentication Code) transported by a *SecurityEncap* option that we discuss later. An URI to the certificate is transported rather then the certificate itself, given the payload restrictions already discussed. When authenticating requestors using public-keys or certificates, the *SecurityToken* option must be sent in a CoAP message also transporting an encrypted MAC (signature). In order to support Kerberos-based authentication domains, a *kerberos* ticket may identify and authorize CoAP requests.

As with the *SecurityOn* option, a CoAP message may transport more than one *SecurityToken* option, thus supporting multiple trust domains and intermediaries.

4.3 *SecurityEncap* CoAP security option

The *SecurityEncap* option transports the security-related data required for the processing of a CoAP message, according to the contents of the *SecurityOn* option. The format of this option is illustrated in Figure 5 and, as for the previous option, only one of the variable-length fields in required for a given CoAP message. The length of this field is thus derived from the length of the option itself.

When providing sender authentication, replay protection and integrity for a CoAP message (in the *SecurityOn* option the *SecurityApplied* field value is 1) this option may be used to transport an encrypted MAC plus a nonce value for freshness. If only encryption is required (the *SecurityApplied* value is 0 in the *SecurityOn* option) this option transports a nonce plus the number of options following in the encrypted part of the payload. As all other options plus the CoAP packet payload are encrypted, the number of options is transported as information helping in the processing of the message by a CoAP intermediary or final entity. In the last scenario the CoAP message is fully protected and all security-related data is transported. The MAC value is computed using the hash or keyed hash algorithm associated with the security context negotiated by the communicating entities and identified in the *SecurityOn* option. The MAC value is computed considering the complete CoAP message plus the options, considering also the *SecurityEncap* option itself with the MAC value field set to all zeros.

4.4 Default security with AES/CCM

The current proposals to standardize security mechanisms for LoWPAN environments and communications are strongly based on the usage of AES/CCM, given its availability at the hardware in wireless sensing platforms supporting IEEE 802.15.4 [2]. Although AES/CCM is available on such platforms to protect messages transmitted at the link-layer, it may also be employed to protect messages of communication protocols at higher layers, by using AES/CCM in the standalone mode. We consider that AES/CCM is the cipher supporting the default CoAP security context, identified with the value 1 and employed when no specific security context has been negotiated. This may be of interest to simple applications employing key pre-configuration or for the initial secure bootstrap of applications employing more complex context negotiation and key management mechanisms. In the default security context AES/CCM is employed with a 12-byte nonce value and an 8-byte MAC. This is in line with the capabilities of current sensing platforms and with the usage of AES/CCM with TLS [13][14], thus enabling the design of cross-layer security mechanisms in the future, for example to support authentication and key management mechanisms for the transport and application-layer. We also consider that applications using the default security context may omit the *Destination Entity* identification on the *SecurityOn* option. This may be appropriate for applications where devices only answer for a default actor URI, while we must note that the final CoAP address is always part of the CoAP request**.**

5 Evaluation of CoAP application-layer message security

Our experimental evaluation allowed us to measure the energetic and computational impact of end-to-end security using CoAP security and DTLS. As our goal is to evaluate end-to-end security in the context of Internet-integrated wireless sensing applications, we consider the usage of a CoAP client residing on an external Internet host and requesting resources from a CoAP server on a LoWPAN wireless sensing device, as illustrated in Figure 6.

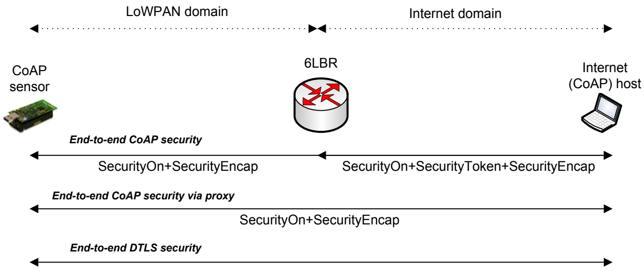

Fig. 6. CoAP and DTLS security end-to-end usage scenarios.

As Figure 6 illustrates, end-to-end security may be achieved in a pure fashion either using DTLS at the transport-layer or the proposed CoAP security options at the application-layer. Alternatively, we also consider the usage of a CoAP intermediary (a forward proxy) in the processing of security. The security intermediary provides authorization of CoAP clients and control of accesses to resources on the LoWPAN via the *SecurityToken* option. We consider the usage of AES/CCM cipher in the default CoAP security context, due on the one side to the availability of this cipher in the TelosB [1] and on the other to guarantee a fair comparison of CoAP security against DTLS as currently proposed for CoAP [5].

5.1 Impact of end-to-end security on CoAP packet payload space

As packet payload space is a scarce resource in LoWPANs environments, our initial evaluation is on the impact of end-to-end security on CoAP packet payload space. Our goal is to analyze if application-layer security leaves enough payload space to transport data from CoAP applications while not requiring costly fragmentations at the 6LoWPAN adaptation layer. Figure 7 illustrates the impact of security on the payload space available for CoAP applications in the presence of end-to-end security. The values illustrated are in percentage of the maximum available payload without security and correspond to the usage scenarios previously illustrated in Figure 6.

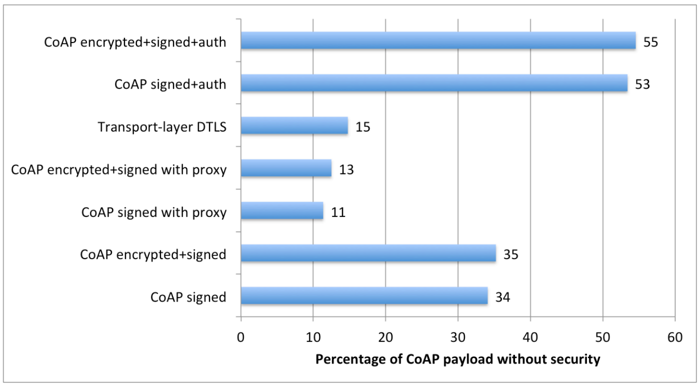

Fig. 7. Impact of end-to-end security on packet payload space available to CoAP.

As we may observe in Figure 7, end-to-end security usage scenarios involving the participation of a CoAP security intermediary (proxy) performs better than DTLS. The usage of a security intermediate thus provides the benefit of permitting the offloading of computationally heavy computations to a more specialized entity while guaranteeing a very small impact on CoAP payload space. The impact of end-to-end security without a proxy on CoAP packet payload space is greater, mostly due to the usage of the *Destination Entity* field in the *SecurityOn* option. We consider that this field requires an average of 20 bytes to transport the URI. Although the impact in this usage scenario is greater, in the worst case 65% of the original 6LoWPAN payload of 88 bytes is still available. Thus, we verify that CoAP security is a viable approach for end-to-end security from the point of view of its impact on packet payload space.

5.2 Impact of end-to-end security on the lifetime of sensing applications

As energy is a critical resource on LoWPAN environments, it directly dictates the lifetime of wireless sensing applications and, in consequence, security mechanisms must be tested against its impact on energy. In our experimental evaluation study we obtained the energy consumption for security using experimental measurements of the voltage across a current sensing resistor placed in series with the battery pack and the circuit board of the TelosB [1]. The energy required for the processing of a 102-byte 6LoWPAN message and related headers (including DTLS and CoAP security headers plus options) was measured as 0.007 nJ (Nano joules). The energy required for the processing of security using AES/CCM in standalone mode for a similar message was measured as 0.2 mJ (Micro joules), while the energy required for the transmission of a packet has been measured as 0.004 nJ (Nano joules) per bit. These experimentally obtained measurements enable us to predict the impact of end-to-end security on the lifetime of CoAP sensing applications.

From the values illustrated in Figure 7 we are able to obtain the maximum payload space that CoAP applications may employ without enforcing costly fragmentation operations at the 6LoWPAN adaptation layer. This corresponds to the usage scenario where end-to-end CoAP security performs encryption, integrity and authentication without the usage of a proxy, for which 45% of the original 6LoWPAN payload (or 40 bytes) is available to transport CoAP data. From this value we subtract 20 bytes required for the transportation of the security-related data (nonce and MAC values) for AES/CCM. Taking into account such considerations and the experimentally obtained values previously discussed we obtain the expectable lifetime for wireless sensing applications in the context of Internet-integrated sensing applications, that we illustrate in Figure 8. We assume the processing and transmission of two messages for each CoAP request, one containing a confirmable request and the other the corresponding reply carrying a piggybacked acknowledgment as defined in CoAP [5]. We also assume the usage of two new AA LR6-type batteries on the TelosB providing a total of 6912 joules of energy.

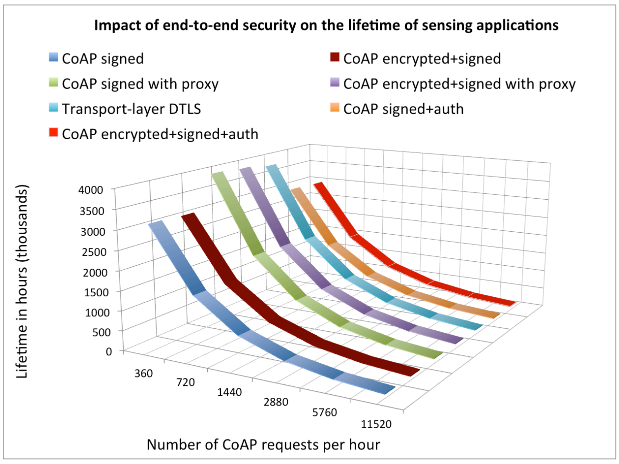

Fig. 8. Impact of end-to-end security on the lifetime of sensing applications.

As in the previous analysis, we observe that end-to-end CoAP security performs better that DTLS when employing a security proxy providing support for the processing of the *SecurityToken* option. Pure end-to-end CoAP security without a security intermediate causes a greater impact on the expected lifetime of sensing applications, particularly for lower communication rates where the cumulative impact of AES/CCM encryption is lower when compared to the impact of the energy required to process and transmit CoAP security options. Despite this observation, the obtained values allow us to conclude that CoAP security provides acceptable lifetime values in all usage scenarios, particularly considering WoT applications designed to require low or moderate wireless communications rates.

As previously discussed, one major motivation of the design of application-layer message security for CoAP is in the support of granular security policies. Security policies may define how each message must be protected, according to the semantics of the CoAP protocol, the type of message, its contents or particular requirements of the application. In this context, our next evaluation considers the following four usage profiles for end-to-end security:

- Applications that only require integrity for CoAP replies containing sensorial data from LoWPAN CoAP devices. In such applications sensorial data is not confidential but must be protected against tampering or communication errors.

- Applications requiring confidentiality and integrity for the same type of CoAP messages. In such applications sensorial data is of sensitive-nature, thus also requiring protection against disclosure.

- Applications requiring confidentiality and integrity but only for CoAP requests transporting authentication-related data using the *SecurityToken* CoAP option. In this case we are concerned with the protection of identity and authorization data against disclosure or tampering.

- Applications requiring confidentiality and integrity for all CoAP messages irrespective of its type or contents. In such applications all messages are considered sensitive from the point of view of security.

In Figure 9 we illustrate the impact of end-to-end security according to the usage profiles previously identified. We are able to clearly observe the advantage of granular security in terms of the lifetime of sensing applications, in comparison with transport-layer DTLS where this approach is unavailable. The only security profile performing worst than DTLS is with CoAP encrypting and signing all messages, due to the difference in terms of the payload space required accommodating security. Despite this, in this scenario the expectable lifetime for applications is large, even considering applications protecting many CoAP messages per hour.

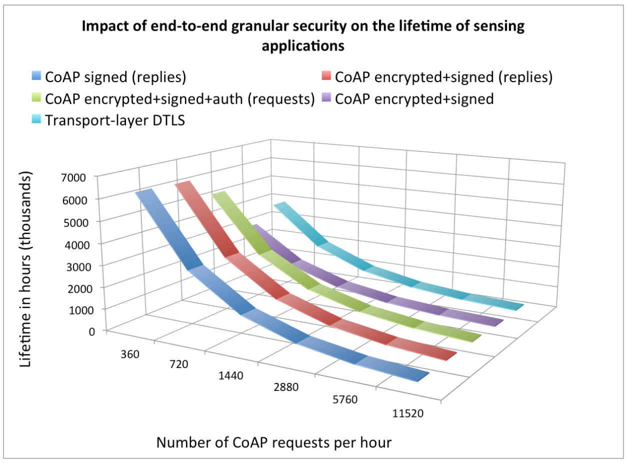

Fig. 9. Impact of (granular) end-to-end security on the lifetime of sensing applications.

Overall, our comparative analysis clearly illustrates the advantages of application-layer message security in protecting CoAP communications. When compared with DTLS, our approach introduces flexibility while providing security functionalities not possible with the transport-layer approach. The usage of security intermediaries participating in security also benefits energy and in consequence the lifetime of sensing applications. We also observe that even when CoAP security is employed to protect all messages as with DTLS, it provides comparable performance.

5.3 Impact of end-to-end security on the communications rate of wireless sensing applications

Our final evaluation is on how CoAP security influences the communications rate achievable by applications. When considering wireless communications using IEEE 802.15.4 at 250Kbit/s, we need to consider the overhead introduced by IEEE 802.15.4 on the bandwidth available for 6LoWPAN and upper protocols, which is of 19.6% of the total bandwidth, given that 25 bytes are required for link-layer information with each 127-byte 6LoWPAN packet. Figure 10 illustrates the maximum transmission rate achievable using DTLS versus the previously described CoAP security profiles. The values illustrated in this Figure are obtained considering our experimental evaluation results and that CoAP transports an average of 20 bytes of payload data per message. We also consider the time required for the application of AES/CCM cryptography to CoAP messages, according to the security usage profiles.

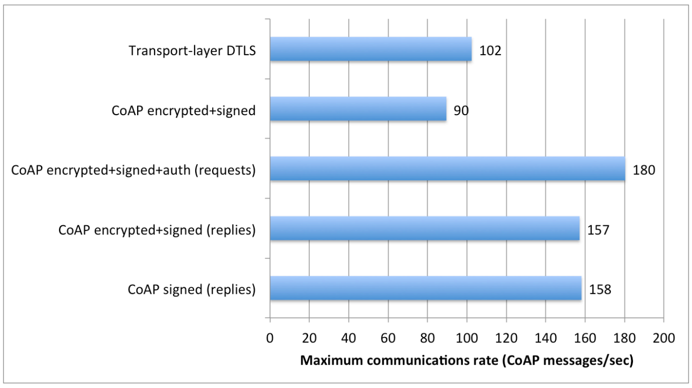

Fig. 10. Impact of end-to-end security on the communications rate of sensing applications.

We may again observe the superior performance of the security profiles requiring the usage of granular application-layer security. CoAP signing and encryption of all messages (as using DTLS) provides inferior performance, but despite this it still allows for 90 CoAP protected messages per second, a limit we may safely consider to be clearly above the requirements of most CoAP wireless sensing applications envisioned for the WoT.

6 Conclusions

The availability of secure end-to-end communications with sensing devices may provide an important contribution to enable WoT wireless sensing applications, as many of such applications may benefit from the availability of direct communications with Internet hosts or external backend servers. Our proposal seeks to provide a contribution in the context of a security architecture supporting Internet-integrated wireless sensing LoWPANs and applications. Our experimental evaluation allowed us to observe that CoAP application-layer security may perform similarly or better than transport-layer security, while supporting functionalities that are not possible with a transport-layer approach. Further research work remains to be done in the context of our proposal, for example in the design of appropriate key management and clock synchronization mechanisms.

Acknowledgements. The work presented in this paper was partly financed by the iCIS project (CENTRO-07-ST24-FEDER-002003) which is co-financed by QREN, in the scope of the Mais Centro Program and European Union's FEDER.

References

1. TelosB Mote Platform, http://www.xbow.com/pdf/Telos_PR.pdf
2. Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE std. 802.15.4, 2006
3. IPv6 over Low power WPAN (6lowpan), https://datatracker.ietf.org/wg/6lowpan/charter/
4. Constrained RESTful Environments (core), https://datatracker.ietf.org/wg/core/charter/
5. Shelby Z. et al. Constrained Application Protocol (CoAP), draft-ietf-core-coap-13, 2013
6. Rescorla E et al. Datagram Transport Layer Security Version 1.2, *RFC 6347*, 2012
7. Gupta V et al. Sizzle: a standards-based end-to-end security architecture for the embedded Internet. *Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom 2005)*, Kauai Island, HI, USA, 2005. DOI: 10.1109/PERCOM.2005.41
8. Jung et al. SSL-based Lightweight Security of IP-based Wireless Sensor Networks. *Proceedings of the International Conference on Advanced Information Networking and Applications Workshop (WAINA '09)*, Bradford, UK, 2009
9. Raza S et al. 6LoWPAN Compressed DTLS for CoAP. *Proceedings of the IEEE 8th International Conference on Distributed Computing in Sensor Systems (DCOSS 2012),* Hangzhou, China, 2012. DOI: 10.1109/DCOSS.2012.55
10. Kothmayr T et al. A DTLS Based End-To-End Security Architecture for the Internet of Things with Two-Way Authentication. *Proceedings of the Seventh IEEE International Workshop on Practical Issues in Building Sensor Network Applications (IEEE SenseApp 2012),* Clearwater, FL, USA, 2012
11. Yegin A, Shelby Z. CoAP Security Options, draft-yegin-coap-security-options-00 (expired April 2012)
12. Hui J et al. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks, *RFC 6282*, 2011
13. Dierks T, Rescorla E. The Transport Layer Security (TLS) Protocol Version 1.2, *RFC 5246*, 2008
14. McGrew D, Beiley D. AES-CCM Cipher Suites for Transport Layer Security (TLS), *RFC 6655*, 2012