2012 IEEE International Conference on Green Computing and Communications, Conference on Internet of Things, and Conference on Cyber, Physical and Social Computing

On the effectiveness of end-to-end security for Internet-integrated sensing applications

Jorge Granjal, Edmundo Monteiro, Jorge Sá Silva
University of Coimbra, Portugal
{jgranjal,edmundo,sasilva}@dei.uc.pt

***Abstract*—While realizing that most of the applications currently envisioned for the Internet of Things (IoT) are critical in respect to security, we may expect that such sensing applications may benefit from the availability of end-to-end IPv6 communications with Internet hosts. Research and standardization work is starting to produce mechanisms that may enable end-to-end communications using IPv6-enabled constrained sensing devices, and such communications will raise serious security challenges that must be addressed in the context of a proper integration architecture that is yet to be standardized for the IoT. In our work we target the fundamental question on the effectiveness of the usage of security mechanisms to protect end-to-end communications involving Internet hosts and constrained sensing devices. We describe mechanisms to enable security at the network-layer and at the application-layer and perform an extensive experimental evaluation study with the goal of identifying the most appropriate secure communications mechanisms and the limitations of current sensing platforms in supporting end-to-end secure communications in the context of Internet-integrated sensing applications.**

Keywords-IoT, security, 6LoWPAN, ESP, AH, CoAP, DTLS

I. INTRODUCTION

Strong security assurances will be required for many applications envisioned for the Internet of Things (IoT) that are expected to process and transmit sensitive data using wireless communications. The fact that the integration of such applications with the Internet may require or benefit from the availability of end-to-end communications between constrained sensing devices and other Internet hosts raises the bar for the security concerns and requirements. Although we realize that a formal layered communications model for the integration of sensing applications with the Internet is yet to be defined, new mechanisms are currently being proposed at the 6LoWPAN (IPv6 over Low Power Personal Area Networks) [1] and Constrained RESTful Environments (CoRE) [2] working groups of the IETF that may enable such end-to-end communications and that therefore are expected to play a major role in the enabling of a future integration architecture for the IoT. In particular, 6LoWPAN targets the design of an adaptation layer to enable the transmission of IPv6 packets over Low-Rate Wireless Personal Area (LoWPAN) networks such as IEEE 802.15.4 [3], while CoRE is currently designing the Constrained Application Protocol (CoAP) [4] to enable Representational State Transfer (RESTful) web communications with constrained sensing devices.

Although 6LoWPAN and CoAP are important proposals in this context, we observe that so far no specific solutions are currently adopted or fully evaluated. Only generic considerations and recommendations [5] have been produced so far for 6LoWPAN, motivating our proposal on the definition of compressed security headers for the adaptation layer. On the other end, three security modes are currently proposed for CoAP [4] that lack an experimental evaluation of its effectiveness using real sensing platforms and applications with particular security requirements. Our goal is therefore to evaluate the effectiveness of the usage of secure end-to-end communications in the context of Internet-integrated sensing applications. We evaluate security at the network-layer against security at the application-layer for such communications with the goal of identifying the most appropriate security mechanisms for applications with particular requirements, while also analyzing if current sensing platforms are able to cope with current proposals for the addressing of security. Our motivation also lies in the fact that end-to-end secure communications with constrained sensing devices may represent an important component of a future secure integration architecture for the IoT, as direct end-to-end communications between sensing devices and Internet or backend hosts may be of benefit to various types of sensing applications envisioned for the IoT.

The paper proceeds as follows. In Sections II and III we describe our proposal on the usage of security at the network-layer and the currently proposed mechanisms to enable security at the application layer. In Section IV such mechanisms are extensively evaluated against critical resources on current sensing platforms. Section V describes an overall evaluation of end-to-end security based on the results from our experimental evaluation study and Section VI concludes the paper.

II. END-TO-END NETWORK-LAYER SECURITY

Applications in areas such as industrial monitoring and control, structural monitoring, home automation, healthcare, vehicle telematics and agricultural monitoring [6] are expected to benefit from end-to-end communications between Internet hosts and constrained sensing devices. Mechanisms are starting to emerge to enable such communications but we currently verify that security is

978-0-7695-4865-4/12 $26.00 © 2012 IEEE
DOI 10.1109/GreenCom.2012.23

currently mostly absent from such proposals. Security mechanisms are thus required to guarantee fundamental security properties such as confidentiality, authentication, integrity and non-repudiation for end-to-end communications. Given that end-to-end security is not a panacea and that security demands for the usage of complementary mechanisms, we must note that mechanisms not addressed in this paper such as key management will also require particular attention from research for the effective support of end-to-end security in the context of Internet-integrated sensing applications. We start by describing our approach to network-layer security and later in the paper discuss the proposed security mechanisms for CoAP.

Considering the lack of proposals for security at 6LoWPAN, we previously proposed and theoretically evaluated two new compressed security headers [7][8] for the adaptation layer that we illustrate in Figures 1 and 2. Our proposal consists on the design of compressed ESP (Encapsulated Security Payload) and AH (Authentication Header) security headers that may enable end-to-end security at the network-layer for communications with constrained IPv6-enabled sensing devices on IEEE 802.15.4 networks. The security headers are identified at the 6LoWPAN adaptation layer using new dispatch type values from the set of reserved values of the original payload byte [9] and the new headers are designed to facilitate its integration into existing implementations of the IP Security architecture [10].

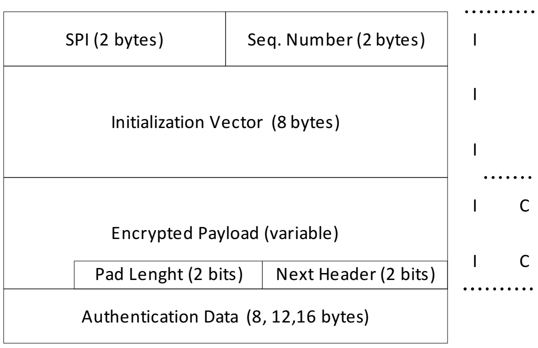
SPI (2 bytes)
Seq. Number (2 bytes)
Initialization Vector (8 bytes)
Encrypted Payload (variable)
Pad Lenght (2 bits)
Next Header (2 bits)
Authentication Data (8, 12,16 bytes)
I
I
I
I C
I C

Figure 1. 6LoWPAN compressed ESP security header

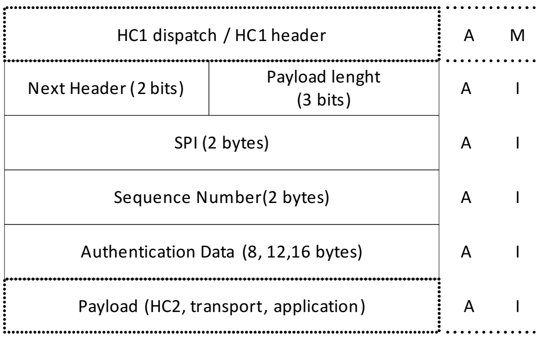
HC1 dispatch / HC1 header
A M
Next Header (2 bits)
Payload lenght (3 bits)
A I
SPI (2 bytes)
A I
Sequence Number(2 bytes)
A I
Authentication Data (8, 12,16 bytes)
A I
Payload (HC2, transport, application)
A I

Figure 2. 6LoWPAN compressed AH security header

In Figure 2 we identify integrity protected fields using an 'I', while encrypted (confidentiality protected) fields are identified with a 'C'. In Figure 3 the parts of the header and data payload that are integrity and authentication protected are indicated by an 'A', while the parts that are considered mutable or immutable in respect to the computation of the Integrity Check Value (ICV) are indicated by an 'M' and 'I', respectively.

Cross-layer optimized security can be implemented by having upper-layers security mechanisms designed to benefit from the availability of hardware security, as we implement for hardware-based AES/CCM. This also facilitates the integration of 6LoWPAN security in the IP Security architecture, as AES/CCM is part of the set of future mandatory algorithms defined for IPSec.

The 6LoWPAN ESP header starts with a 2-byte Security Parameters Index (SPI) field, followed by a 2-byte sequence number and an 8-byte Initialization Vector (IV). The IV is compatible with all the current and future mandatory cryptographic algorithms of the IP Security architecture, and also guarantees compatibility with current and future cryptographic suites based on AES. Next in the packet comes the encrypted data, at the end of which comes the pad length and header length fields. The ESP header at the end uses an ICV or Message Integrity Code (MIC) field. This field also guarantees compatibility with current and future security suites of the IP Security architecture. The AH header starts with a next header field, followed by the payload length field storing the total length of the header in units of 32-bit words. The remaining fields are used with a purpose similar to compressed ESP.

In our experimental evaluation later in the paper we consider only the usage of transport mode network-layer security, as it is clearly the most useful mode given the limitations of payload space on 6LoWPAN networks. This does not preclude however the usage of tunnel mode security in scenarios where it is found to be useful, for example when 6LoWPAN is able to efficiently compress the packet header and consequently leave more space for security and application data, for instance when two sensing devices on different LOWPANs communicate via a 6LoWPAN security gateway.

III. End-to-end application-layer security

CoAP web communications may be secured using DTLS [11] over UDP, and the security modes proposed for CoAP are currently identified as the *PreSharedKey*, *RawPublicKey* and *Certificates* modes. The current proposal is for the *RawPublicKey* and *Certificates* security modes to use ECC (Elliptic Curve Cryptography) to support authentication of devices and messages using ECDSA (Elliptic Curve Digital Signature Algorithm) [12]. In a similar vein, key agreement is supported by ECDH (Elliptic Curve Diffie-Hellman) [12]. The *Certificates* mode also enables the alternative usage of SHA to support integrity. The fact that ECC cryptography may be too resource demanding for constrained sensing devices also contributes to our interest in the experimental evaluation of end-to-end application-layer security against the alternative proposal of network-layer security. The three security modes for CoAP target deployment scenarios of sensing applications with different characteristics, as we proceed to discuss.

The *PreSharedKey* security mode targets deployment scenarios where devices store predefined keys used in communications with other devices or with group of other devices and without using public-key cryptography. This will be the case for example in deployment scenarios where devices may be preconfigured with security and other management data and operate in an unattended fashion without a security infrastructure. The fact the public-key security is also not required in this mode may make it more appropriate for very constrained sensing devices. This mode employs the TLS_PSK_WITH_AES_128_CCM cipher suite in the AEAD (Authenticated Encryption with Associated Data) [13] operational mode AEAD_AES_128_CCM [14], using a 128-bit authentication tags and a 12-byte nonce with each packet protected using the same cryptographic key. Integrity is supported using the Pseudorandom Function (PRF) defined for TLS 1.2 [14] and HMAC with SHA-256.

In the *RawPublicKey* mode a sensing device possesses one or more public keys from which it derives its identification and which it uses to authenticate other communication parties, although a certification chain is not used. This security mode is therefore intended for devices and applications that are able to support ECC public-key cryptography while not requiring the usage of a certification infrastructure. This security mode employs the TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8 security suite that uses Elliptic Curve Diffie-Hellman with Ephemeral Keying (ECDHE) and ECDSA ECC. As in the previous security mode, it uses the AEAD_AES_128_CCM [14] operational mode.

The *Certificates* mode borrows many operational aspects from *RawPublicKey* but in this case a device must be able to authenticate its peer using public keys obtained from certificates. Therefore, this mode is intended for deployments where sensing devices are able to support public-key cryptography and a security infrastructure is available so that a device is able to use root trust anchors for certificate validation. Given that devices may be unable to support ECC cryptography, this security mode also enables the usage of the security suite TLS_RSA_PSK_WITH_AES_128_CBC_SHA that uses pre-shared keys between client and servers and employs RSA for device authentication and key-agreement using RSA_PSK [15]. With this security suite confidentiality is guaranteed using AES in CBC mode and integrity with SHA. Now that we have described the proposed mechanisms intended to secure end-to-end communications with constrained sensing devices, we proceed by discussing our experimental evaluation study on its usage.

IV. Experimental evaluation of end-to-end Network-layer and application-layer security

As for any proposal regarding the usage of computationally-demanding security mechanisms for resource-constrained sensing devices, the effectiveness of end-to-end security should be determined by evaluating the impact of the proposed mechanisms using real sensing platforms and considering requirements from particular sensing applications. The experimental evaluation of new mechanisms is of particular importance when dealing with constrained sensing devices, as in practice several unpredicted aspects related to the operations of such devices and wireless communications are difficult to reproduce realistically using simulation environments. The experimental evaluation study we describe next will enable the identification of the impact of the described approaches for end-to-end security on constrained sensing platforms, and the results from this study will form the ground for our later overall evaluation of the effectiveness of end-to-end security for Internet-integrated sensing applications.

A. Experimental evaluation setup

Our experimental evaluation employs end-to-end 6LoWPAN/CoAP communications between a TelosB [16] mote and a Linux host. The TelosB runs the TinyOS [17] operating system and supports 6LoWPAN, CoAP and security. The Linux host performs routing between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN by employing a second TelosB mote as a bridge.

TABLE I
SECURITY CONFIGURATIONS FOR END-TO-END COMMUNICATIONS

Cryptographic suite or Operational mode	Usage	Security provided
3DES-CBC	6LoWPAN ESP	Confidentiality
AES-XCBC-MAC-96		Integrity, authentication
3DES-CBC	6LoWPAN ESP	Confidentiality
HMAC-SHA1-96		Integrity, authentication
AES-CBC	6LoWPAN ESP	Confidentiality
AES-XCBC-MAC-96		Integrity, authentication
AES-CBC	6LoWPAN ESP	Confidentiality
HMAC-SHA1-96		Integrity, authentication
AES/CCM (HW)	6LoWPAN ESP	Confidentiality, integrity, authentication
AES-XCBC-MAC-96	6LoWPAN AH	Integrity, authentication
HMAC-SHA1-96	6LoWPAN AH	Integrity, authentication
AES/CCM (HW)	6LoWPAN AH	Integrity, authentication
TLS_PSK_ WITH_AES_128_CCM_8	CoAP with DTLS	Confidentiality
TLS 1.2 PRF (SHA-256)		Integrity, authentication
TLS_ECDHE_ECDSA_ WITH_AES_128_CCM_8	CoAP with DTLS	Confidentiality
TLS 1.2 PRF (SHA-256)		Integrity, authentication
TLS_RSA_PSK_ WITH_AES_128_CBC_SHA	CoAP with DTLS	Confidentiality
TLS 1.2 PRF (SHA-1)		Integrity, authentication

Although our experimental results are particular to the TelosB, this platform is currently considered to be a good representative of the currently available sensing platforms, therefore providing an appropriate reference. The TelosB is powered by a 16-bit RISC MSP 430 microcontroller with 48Kbytes of ROM and 10Kbytes of RAM, supporting communications at 2.4GHz and data transmissions at

250Kbps. Given its support of IEEE 802.15.4, it also provides AES/CCM hardware cryptography that we include in our implementation and evaluation as an important cross-layer security optimization benefiting both 6LoWPAN and CoAP security. Given that end-to-end security may require the simultaneous usage of more than one security suite, it is important to begin by clearly defining how the proposed and evaluated mechanisms are employed, as described in Table I.

For the security suites requiring AES in CCM mode we always use hardware cryptography, while the remaining algorithms are supported using code optimized for small microcontrollers with the characteristics of the MSP 430. Each algorithm is used with its inherent cryptographic block and key size, also in line with the configurations required by the IP Security architecture and CoAP. As hardware-based AES/CCM cryptography still requires software support, we employed the standalone hardware encryption code from the Shanghai Jiao Tong University [18] for this purpose. ECC is supported using TinyECC [19], while RSA, AES/CBC and SHA-1 are evaluated using code optimized for 8-bit architectures.

B. Experimental evaluation results

Our experimental evaluation study targets the measurement of the impact of network-layer and application-layer security on resources that are critical on battery-powered sensing platforms with the characteristics of the TelosB, namely memory, energy and computational time. The usage of such resources dictates the usefulness of any proposal for constrained sensing devices, as it directly influences fundamental aspects such as the lifetime of sensing applications or the rate at which sensing devices are able to communicate.

1. Memory footprint of end-to-end security

Our experimental evaluation study measures the RAM and ROM memory necessary with each version of a TinyOS testing applications supporting 6LoWPAN and the various security configurations described in Table I. In Figure 3 we illustrate the memory requirements of each end-to-end security configuration. Memory usage is represented in percentage of the total of RAM and ROM memory on the TelosB and, for comparison purposes, we also illustrate the memory required for two base usage scenarios, one using a TinyOS application with BLIP and CoAP support without security, and the other a TinyOS application with BLIP and 6LoWPAN security headers but without any cryptographic algorithm.

We observe that hardware-level encryption doesn't come without a non-negligible overhead on memory, particularly in terms of ROM memory. We can observe the limitations of the TelosB regarding memory, as not enough ROM memory is available to support all cryptographic operations required for CoAP security using the security mode TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, as it requires the simultaneously support of ECCDH, ECCDSA and AES/CCM. RAM may also be a problem in usage scenarios where larger applications than our test applications are required to run on the mote. Application-layer security presents in general a larger impact on memory when compared with network-layer security, with the exception of network-layer security using 3DES, however with the disadvantage of being of less interest and usable only as a last resort in situations where AES is unavailable, considering its superior security.

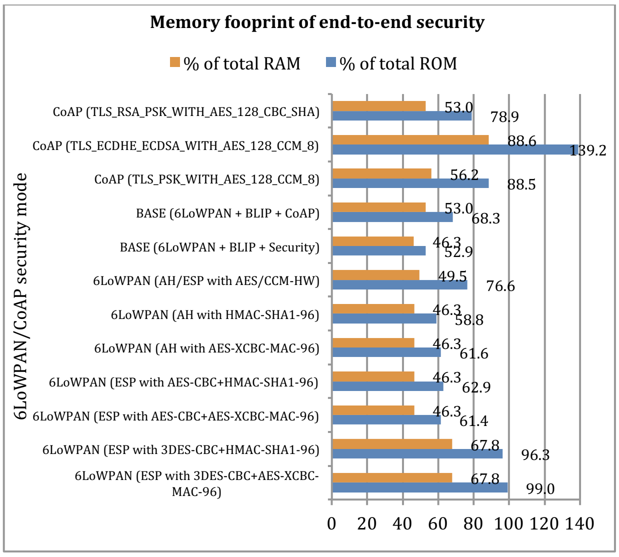

Figure 3. Memory footprint of end-to-end security

Other than network-layer security using 3DES and application-layer security using ECC, we observe that the remaining security modes can be considered viable in respect to its requirements on memory. Considering the representativeness of the TelosB, we are able to conclude that sensing platforms should evolve to support more memory space, so that network-layer and application-layer security is supported while leaving a safe margin of available memory space to appropriately support other required applications on the mote.

2. Computational and energy overhead of end-to-end security

The computational time required to process security for a given packet directly influences the maximum communications rate that a smart object is able to achieve. Energy is also a scarce resource on sensing platforms, as many sensing applications are designed for battery-powered sensing devices and to run for extended periods of time. Our study proceeds by analyzing the impact of end-to-end security on these important resources. In Figures 4 and 5 we illustrate the experimentally obtained values for the energy and computational time required to process security for a fully sized 102-byte 6LoWPAN packet with the security configurations previously described in Table I. We employ a logarithmic scale due the large range of values. Energy was obtained 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. Computational time was measured using the 32 KHz internal oscillator of the TelosB. We do not distinguish MIC codes with different sizes, due to the fact that AES-XCBC-MAC-96 and HMAC-SHA1-96 always generate 12-byte MIC codes, while for hardware AES/CCM the energy required for the generation of a 16, 12 or 8 bytes MIC using standalone hardware encryption is in practice the same.

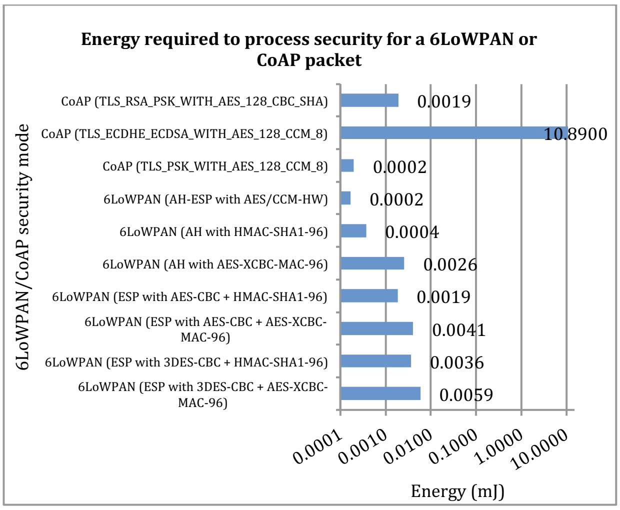

Figure 4. Energy required for end-to-end security

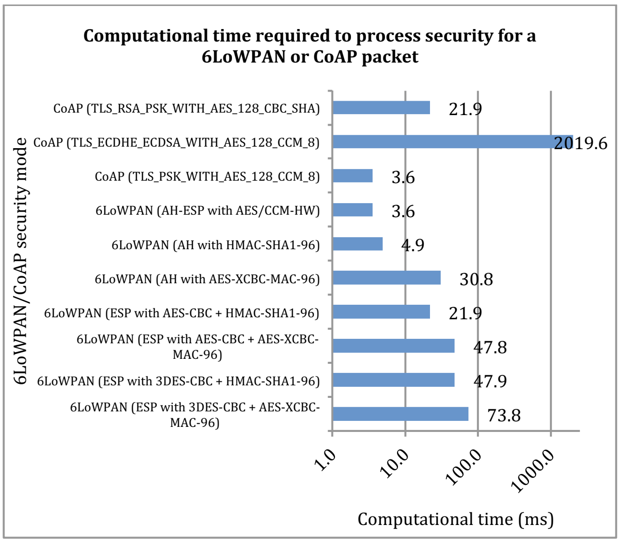

Figure 5. Computational time required for end-to-end security

The illustrated values already include the energy and computational time required for the processing of 6LoWPAN and CoAP headers, including security, in the context of the BLIP networking stack. These values are total, measured from the reception of a 6LoWPAN/CoAP packet to the time when the respective cryptographic algorithm finishes processing the packet, and therefore represents the total computational and energetic effort required to process security in the context of network-layer or application-layer communications.

We clearly observe that ECC public-key cryptography is very demanding and may be considered viable only for applications requiring very low transmission rates. In particular, TLS_ECDHE_ECDSA_WITH_AES_CCM_8 processes each packet with AES/CCM and ECDSA, with ECDH only being considered for key establishment purposes during the establishment phase of a DTLS session. Although ECC-based cryptography is a good alternative to classical public cryptography, it still represents a bottleneck using current sensing devices. The remaining CoAP security modes are much more efficient in terms of its impact on energy and computational time. The security suite TLS_PSK_WITH_AES_128_CCM_8 is the most efficient mode to protect application-layer communications with DTLS, as expected due to its usage of AES/CCM. TLS_RSA_PSK_WITH_AES_128_SHA on the other end provides a good alternative to ECC when public-key authentication is required. Network-layer security in general presents a moderately greater impact when compared with application-layer security, with the exception of 6LoWPAN with AH and ESP using AES/CCM and of AH using the extremely efficient HMAC-SHA1-96. With the exception of ECC-based security for the application-layer, we observe that end-to-end security can be considered viable in respect to its impact on energy and computational time.

V. Overall Evaluation of End-to-End Security for Internet-Integrated Sensing Applications

We now proceed by using the results from our previously described experimental evaluation study in an overall evaluation on the effectiveness of end-to-end security considering its usage with sensing applications with particular security usage requirements and deployment characteristics. Our goal is to evaluate the applicability of end-to-end security for such applications, while identifying the most effective approach for each usage scenario.

A. Security usage profiles for Internet-integrated sensing applications

Given the diverse application areas envisioned for the IoT, we are able to characterize a set of representative applications in respect to its fundamental requirements on security, along with the security modes that may be used to enable either network-layer or application-layer security.

As Table II illustrates, applications are differentiated by its requirements in terms of the integration with existing public-key infrastructures and support of fundamental security properties. For some applications authentication and integrity is sufficient, while others may also require confidentiality for end-to-end communications. Given the results from our experimental evaluation study, we consider the usage of network-layer security with ESP using AES/CCM or in alternative of AH using SHA1.

TABLE II
SECURITY USAGE PROFILES FOR INTERNET-INTEGRATED SENSING APPLICATIONS

Confidentiality	Authentication and integrity	Support of web services	Public-key infrastructure	Application areas	Security usage modes
Yes	Yes	No	No	Industrial control and monitoring	6LoWPAN ESP in all modes
No	Yes	No	No	Structural/ agricultural monitoring	6LoWPAN AH in all modes
Yes	Yes	Yes	Yes	Healthcare, vehicular applications	CoAP with TLS_RSA_WITH_AES_128_CBC_SHA or TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8
Yes	Yes	Yes	No	Home automation	CoAP with TLS_PSK_WITH_AES_128_CCM_8

As for application-layer security, we consider the usage of ECC or RSA for applications requiring public-key cryptography. For the home automation application area, we also consider that security keys and other required data are preconfigured on sensing devices. The classification in Table II provides the ground for our following analysis on the impact of end-to-end security in the context of particular Internet-integrated sensing applications.

B. Impact of end-to-end security on the communications rate of sensing devices

In Table III we describe the maximum transmission rate achievable using end-to-end security, with the values being valid for applications requiring secured transmissions of packets measuring at most 54 bytes in order to avoid fragmentation. When considering communications using IEEE 802.15.4 at 250Kbit/s, we cannot exclude from consideration the overhead introduced by IEEE 802.15.4 on the bandwidth available for 6LoWPAN and application data. This overhead represents 19.6% of the total bandwidth, as 25 bytes are required for link-layer information with each 127-byte 6LoWPAN packet. We also consider the time required for the processing of 6LoPWAN and other (network-layer or CoAP) required headers, which we have experimentally measured as 0.09 milliseconds on the TelosB.

TABLE III
MAXIMUM TRANSMISSION RATES USING END-TO-END SECURITY

Security mode/Cipher suite	Maximum transmission rate (packets/sec)
CoAP using TLS_PSK_WITH_AES_128_CCM_8	132.1
CoAP using TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8	0.49
CoAP using TLS_RSA_PSK_WITH_AES_128_CBC_SHA	38.7
6LoWPAN ESP/AH using AES/CCM (HW)	132.1
6LoWPAN AH using HMAC-SHA1-96	112.7
6LoWPAN AH using AES-XCBC-MAC-96	28.7
6LoWPAN ESP using AES-CBC and HMAC-SHA1-96	38.6
6LoWPAN ESP using AES-CBC and AES-XCBC-MAC-96	19.3
6LoWPAN ESP using 3DES-CBC and HMAC-SHA1-96	19.6
6LoWPAN ESP using 3DES-CBC and AES-XCBC-MAC-96	12.8

Table III does not represent the maximum transmission rates for network-layer communications without security, which is of 252 packets/sec, and for application-layer communications without security, which is of 246 packets/sec. We again observe the impact of ECC, implying that ECC-based end-to-end security may only be viable for sensing applications requiring low transmission rates or targeting short-term deployments. Nevertheless, it is possible that roughly one packet transmitted every 2 seconds may be sufficient for particular sensing applications. We observe that applications in industrial control and monitoring using network-layer security may preferably use ESP with AES/CCM. For sensing platforms without hardware-based AES/CCM, network-layer security may be implemented using ESP with HMAC-SHA1-96 and AES-CBC. Applications requiring only integrity and authentication for network-layer communications, such as in structural and agricultural monitoring areas, may preferably employ AH with HMAC-SHA1-96. We find HMAC-SHA1-96 to be very efficient and therefore an excellent alternative to provide secure hashing in platforms not supporting hardware-based AES/CCM. Sensing applications requiring the support of web communications without public-key cryptography may use the very efficient security mode TLS_PSK_WITH_AES_128_CCM_8. RSA represents a good alternative to ECC if public-key cryptography is required for application-layer security. We also observe that, with the exception of ECC-based application-layer security, acceptable end-to-end communication rates can be obtained with security both at the network and application layers.

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

Most sensing applications designed for the IoT will probably only be viable if able to operate in unattended mode during an acceptable period of time, as in many deployments sensing devices are required to use batteries. In Figure 6 we illustrate the achievable lifetimes for Internet-integrated sensing applications using end-to-end security, considering communications at the maximum transmission rates previously described in Table III.

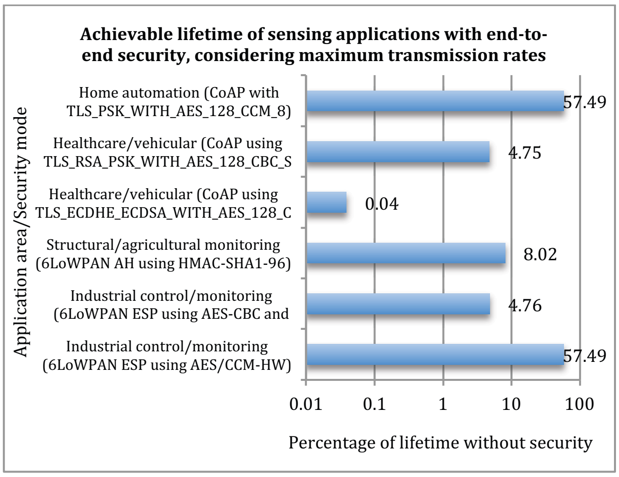

Figure 6. Lifetime of sensing applications with end-to-end security

The values are in percentage of the total lifetime achievable without security, which is of 2338 days, and were obtained considering a TelosB sensing device powered using two new AA LR6-type batteries. We also consider the energy required for the processing of 6LoWPAN and CoAP security headers, which we experimentally measured as 0.007 Nano joules (nJ) per processed packet. This value reflects the total energy required for the processing of a packet. The larger impact of ECC is again clearly visible on the lifetime expectable for sensing applications using application-layer security with the security suite TLS_ECDHE_ECDSA_WITH_AES_128_CCM_8, as it provides less than one day for applications requiring transmissions at the maximum rate. For applications designed to require lower transmission rates, the lifetime using ECC may be significantly improved, for example up to 26 days for applications requiring as much as one packet per minute.

Despite the efficiency of AES/CCM, we also observe its impact on the lifetime of the device. Network-layer and application-layer end-to-end security find their most viable usage scenario when employing hardware-based AES cryptography, and even so a significant impact is visible on the lifetime of applications, in particular 111 days for industrial monitoring applications employing network-layer security using ESP with AES-CBC and HMAC-SHA1-96, and 187 days for structural and agricultural monitoring applications employing network-layer security using AH with HMAC-SHA1-96. For application-layer security and other than when using ECC-based security, TLS_RSA_PSK_WITH_AES_128_CCM_8 provides the worst usage scenario with 111.1 days for healthcare/vehicular applications. We must nevertheless observe that Figure 6 illustrates worst-case values, as the represented values are obtained considering communications at the maximum achievable rate.

We may fairly observe that for the security usage profiles described in Table II end-to-end security still provides acceptable lifetime for the respective applications, as long as such applications require moderate or low transmission rates. Certainly, lifetime values may be improved for applications requiring lower transmission rates or higher transmission rates mostly in shorts bursts.

VI. Conclusions

We have proposed, described and evaluated the impact of mechanisms to secure end-to-end communications with sensing devices in the context of Internet-integrated sensing applications. On the one side we have observed that, as long as applications are able to accept compromises between security, communications rate and resources usage, end-to-end security is indeed viable at the network and application layers. Network-layer security provides the benefit of enabling end-to-end secure communications irrespective of the applications, while on the other end application-layer security may facilitate the integration with certification infrastructures via the usage of ECC public-key cryptography, even if at the expense of more resources of the sensing platform. Both approaches are viable and valuable and can be employed in the context of a flexible and adaptable secure integration architecture for the IoT. Adaptability in this context implies that applications should be able to select the most appropriate security mode according to its requirements. Limitations of current sensing platforms were also identified and should be targeted in future designs to facilitate the secure integration of sensing applications with the Internet, particularly the support of more RAM and ROM memory and the integration of ECC at the hardware.

References

[1] IPv6 over Low power WPAN (6lowpan), https://datatracker.ietf.org/wg/6lowpan/charter/ (accessed July 2012).

[2] Constrained RESTful Environments (core), https://datatracker.ietf.org/wg/core/charter/ (accessed July 2012).

[3] Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE std. 802.15.4, 2006.

[4] Shelby Z. et al. Constrained Application Protocol (CoAP). draft-ietf-core-coap-11, 2012.

[5] Park S, Kim K, Haddad W, Chakrabarti S, Laganier J. IPv6 over Low Power WPAN Security Analysis. draft-daniel-6lowpan-security-analysis-05, 2011.

[6] Kim E. et al. Application Spaces for 6LoWPANs. draft-ietf-6lowpan-usecases-10, 2011.

[7] Granjal J, Silva R, Monteiro E, Silva JS, Boavida F. Why is IPSec a viable option for wireless sensor networks. Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems (MASS 2008), Atlanta, USA, 2008. DOI: 10.1109/MAHSS.2008.4660130.

[8] Granjal J, Monteiro E, Silva JS. Enabling Network-Layer Security on IPv6 Wireless Sensor Networks. Proceedings of IEEE GLOBECOM 2010, Miami, USA, 2010. DOI: 10.1109/GLOCOM.2010.5684293.

[9] Montenegro G. et al. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, 2007.

[10] Kent S, Seo K. Security Architecture for the Internet Protocol. RFC 4301, 2005

[11] Rescorla E., Modadugu N. Datagram Transport Layer Security Version 1.2. RFC 6347, Jan 2012.

[12] SECG-Elliptic Curve Cryptography-SEC 1, http://www.secg.org (accessed July 2012).

[13] McGrew D. An Interface and Algorithms for Authenticated Encryption. RFC5116, January 2008.

[14] McGrew D. et al. AES-CCM ECC Cipher Suites for TLS. draft-mcgrew-tls-aes-ccm-ecc-00, 2011.

[15] Eronen P., Tschofenig H. Pre-Shared Key Cipher Suites for Transport Layer Security (TLS). RFC 4279, 2005.

[16] TelosB Mote Platform, http://www.xbow.com/pdf/Telos_PR.pdf (accessed July 2012).

[17] TinyOS Operating System, http://www.tinyos.net/ (accessed July 2012).

[18] Standalone hardware AES Encryption using CC2420, http://cis.sjtu.edu.cn/index.php/The_Standalone_AES_Encryption_of_CC2420_(TinyOS_2.10_and_MICAz (accessed July 2012).

[19] Liu A., Ning P. TinyECC: A Configurable Library for Elliptic Curve Cryptography in Wireless Sensor Networks. Proceedings of the 7th h international conference on Information processing in sensor networks (IPSN '08), 2008.