INTERNATIONAL JOURNAL OF COMMUNICATION SYSTEMS
Int. J. Commun. Syst. (2012)
Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/dac.2444

Network-layer security for the Internet of Things using TinyOS and BLIP

Jorge Granjal*,†, Edmundo Monteiro and Jorge Sá Silva

Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal

SUMMARY

The design of standard communications and security mechanisms for resource-constrained sensing applications and devices may provide an important contribution for its integration with the Internet and consequently towards the realization of what we nowadays identify as the Internet of Things. This vision will only be realizable if appropriate security mechanisms are available, and in this context we target the design and experimental evaluation of security mechanisms for communications at the network-layer with sensing devices (smart objects) using the standard IPv6 protocol. Our work proposes and evaluates the usage of new compressed security headers for the network layer with smart objects. We implement and evaluate what is, as far as we know, the first proposal of security at the network layer experimentally evaluated using the TinyOS operating system and its networking stack. As we verify in the course of our evaluation study, various scenarios employing network-layer secure communications involving smart objects are feasible, particularly when security mechanisms are designed to benefit from cross-layer interactions that allow the optimization of expensive cryptographic operations. Copyright © 2012 John Wiley & Sons, Ltd.

Received 15 August 2011; Revised 5 August 2012; Accepted 9 September 2012

KEY WORDS: Internet of Things; smart objects; 6LoWPAN; compressed security headers; TinyOS; BLIP

1. INTRODUCTION

Strong security assurances will be required for many applications on the Internet of Things (IoT) that are expected to process and transmit sensitive data using wireless communications [1, 2]. Security mechanisms should thus be designed and adopted for the IoT that are both standard and flexible. Standardization enables the widespread adoption of compatible security solutions [3], while flexibility may guarantee that security mechanisms may be easily adapted to a wide range of heterogeneous sensing devices and applications. Although it is certain that not all smart objects on the IoT will have the capability or be required to support IPv6, the availability of secure end-to-end communications at the network layer with other sensing devices or with Internet hosts may enable a much richer integration of sensing applications with the Internet. It may also enable new types of sensing applications where smart objects are able to cooperate remotely and securely using Internet communications.

The 6LoWPAN (IPv6 over Low Power Personal Area Networks) group of the Internet Engineering Task Force (IETF) was mandated with the task of designing an adaptation layer that enables the transmission of IPv6 packets over IEEE 802.15.4 [4] networks. Despite this initial focus on a particular technology, other communications standards such as Bluetooth Low Energy or Power Line Communication are expected to be supported in the future, allowing a myriad of heterogeneous sensing and actuating devices to communicate using standard Internet Protocols (IPs) [5, 6]. This aspect may certainly contribute to the evolution of the current Internet architecture towards

*Correspondence to: Jorge Granjal, Department of Informatics Engineering, University of Coimbra, Coimbra, Portugal.
†E-mail: jgranjal@dei.uc.pt

something we currently identify as the IoT, an Internet where communications with sensing devices and applications are supported and transparent. Other than the adaptation layer, the 6LoWPAN group has also defined mechanisms such as neighbor discovery and address auto-configuration that allow a sensing device to activate its presence on an existing IPv6 network of smart objects. The 6LoWPAN group has produced the RFC 4919 [7] discussing general goals and assumptions of the group and the RFC 4944 [8] describing the adaptation layer and related header compression mechanisms. As we analyze throughout the paper, header compression is omnipresent in all 6LoWPAN solutions, given the extremely limited payload space to transmit data using LoWPAN technologies such as IEEE 802.15.4.

Although the successful integration of 6LoWPAN networks with the Internet will require security to be properly addressed from the start [9], we note that it has not been properly addressed in 6LoWPAN, because only generic considerations and recommendations [10] have been produced so far, but not any specific mechanism to enable security in the context of the adaptation layer. There is therefore no current solution to enable secure end-to-end communications with IPv6-enabled smart objects using the adaptation layer, probably because of the assumption that security will be addressed in other layers. We must realize that in practice 6LoWPAN enables many interesting usage scenarios, for example with two smart objects on remote locations communicating in the context of a distributed sensing application, or when an Internet host is able to obtain information directly from a sensing device. Therefore, because network-layer security was designed as a cornerstone of the current Internet, it may also play an important part in the context of a broader secure integration architecture for the IoT.

We previously pioneered the idea of enabling security at the network layer for 6LoWPAN smart objects [11]. More recently, we proposed a secure interconnection model [12, 13] where such mechanisms are theoretically validated. In the current paper we describe the implementation and the experimental evaluation of such mechanisms, considering its requirements of vital resources on resource-constrained smart objects.

The paper proceeds as follows. In Section 2 we analyze related work, and Section 3 shows that the proposed compressed 6LoWPAN security headers can be employed. Section 4 describes the experimental evaluation setup used to validate our proposal and in Section 5 we analyze the results obtained from the experimental evaluation study. In Section 6 we analyze the overall effectiveness of 6LoWPAN security, and Section 7 concludes the paper.

2. RELATED WORK

Current proposals to implement secure end-to-end communications between smart objects and Internet hosts mostly target the transport layer, in particular by proposing modified versions of the SSL (Secure Sockets Layer) protocol. For example SSNAIL [14] proposes a light-weight version of SSL to be supported by Internet hosts and smart objects. Other proposals do exist that only provide partial end-to-end security such as Sizzle [15], which employs SSL to secure communications between an Internet host and a security gateway protecting the network of smart objects from the Internet, with such communications being translated to a proprietary communications protocol in the network of smart objects. Another research proposal can be found in ContikiSec [16], which, although providing security at the link layer only, introduces the idea of employing different security modes that can be related to the security requirements of a particular sensing application or device. Table I illustrates the main characteristics of these research proposals.

These proposals have shown that security can be effectively employed at higher communication layers with resource constrained smart objects, something that is in deep contrast with the classic perception of many researchers. Nevertheless, two important aspects are missing from these proposals that we believe are vital for security in the context of the IoT, and can be (at least partially) answered by network-layer security. One is that security mechanisms should be available that provide security for communications independently of the applications. In this respect, SSL presents the limitation of requiring explicit support from sensing applications. Another relevant aspect is that security mechanisms should be adaptable to the characteristics and security requirements of

DOI: 10.1002/dac

Table I. Research proposals for security at high layers with smart objects.

	SSNAIL	Sizzle	ContikiSec
Authentication	ECC (ECDSA)	ECC (ECDSA)	CMAC
Key negotiation	ECC (ECDH)	ECC (ECDH)	Not supported
Key size(s)	160 bits	160 bits	128 bits
Data encryption	RC4	RC4	AES
Hashing/Integrity	MD5, SHA1	MD5, SHA1	CMAC
Access control	Not supported	Security gateway	Not supported
Operational layer	Transport (SSL)	Transport (SSL)	Link-layer
Gateway usage	No	Yes	No
End-to-end security	Yes, with SSL	Yes, with SSL	Not supported

particular sensing applications. Regarding this aspect, mechanisms that work with fixed configurations in terms of parameters that control its security and resource usage may be limitative for the IoT. Aspects such as the cryptographic algorithms employed and relevant configuration parameters such as cryptographic key size and frequency of key refreshment deeply influence the lifetime of sensing applications and resource-constrained devices. Security mechanisms should therefore allow the establishment of acceptable compromises between resources required for performing security operations and the security level required for a particular sensing application. Other than the proposals described in Table I, another proposal also exists to enable security at the 6LoWPAN layer [17] with the same goal as ours. This proposal is implemented in Contiki [18], but its validation suffers from various limitations that we strive to address in the current work. Authors do not consider the usage of security in tunnel and transport modes when evaluating its impact on payload space, neither the usage of variable-sized keys and authentication data, two important requirements for the adaptability of security to applications with different requirements in terms of security. Also, a study on the impact of the proposal on the lifetime of sensing applications is not available. Finally, energy consumption is estimated using Contiki's integrated energy estimator [17], but we believe that experimental measurements using real sensing platforms allow us to clearly investigate the effectiveness of new proposals for constrained sensing platforms.

It is our belief that security can be integrated at the 6LoWPAN adaptation layer with the characteristics previously identified as desirable, and thus enabling the usage of application-independent and flexible security mechanisms, which can play an important part in the integration of smart objects with the Internet. Security at the network-layer is certainly not a solution to all security issues and extremely restricted devices such as radiofrequency identification devices may require different approaches to security [19, 20]. This implies that the security architecture adopted for the IoT must also target and integrate other required security mechanisms.

3. SECURITY IN THE 6LOWPAN ADAPTATION LAYER

We begin by discussing security in the context of 6LoWPAN header compression mechanisms, a fundamental concept of the adaptation layer and that must therefore be considered during the design of new security mechanisms.

3.1. Security in the context of header compression

One major goal of the 6LoWPAN adaptation layer is the support of fragmentation and reassembly of IPv6 packets transmitted over LoWPANs. This is a necessity because IPv6 determines that any communications link may be able to support a minimum maximum transmission unit (MTU) of 1280 bytes, while the MTU of LoWPANs is typically lower. For example, with IEEE 802.15.4 only 102 bytes are available (without link-layer security) of payload space, as Figure 1 illustrates. The payload space available when using IEEE 802.15.4 depends on the overhead introduced by addressing and control information at the link layer. Because IEEE 802.15.4 also provides security at the link layer [4], its usage also influences the payload space available at the end.

DOI: 10.1002/dac

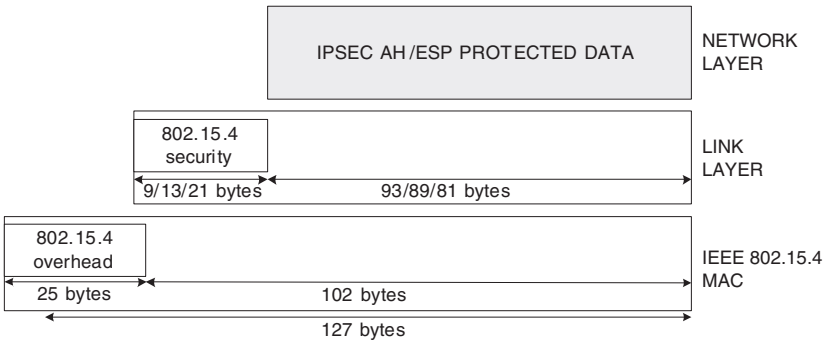

Figure 1. Payload space available for 6LoWPAN using IEEE 802.15.4 considering also the usage of link-layer security.

As illustrated in Figure 1, IEEE 802.15.4 provides three link-layer security modes. The AES-CCM-128 security mode requires 21 bytes of payload space, AESCCM64 requires 13 bytes and AES-CCM-32 requires 9 bytes. We are considering the usage of an IEEE 802.15.4 auxiliary security header occupying 5 bytes, with 1 byte being used for the security control field and 4 bytes for the frame counter field, also considering that the cryptographic keys required for security are obtained automatically from the source and destination link-layer addresses of the frame [4]. Because network-layer security can protect communications even for data transmitted in a wireless network of smart objects, for the purpose of the evaluation of our proposal later in the paper, we consider the availability of 102 bytes as the data payload for 6LoWPAN, meaning that we dispense link-layer security. Because link-layer security is available at the hardware in many sensing devices, the fact that link-layer security mechanisms are not activated does not mean that such efficient encryption and authentication mechanisms cannot be of use. We consider the design of cross-layer security mechanisms for the 6LoWPAN adaptation layer, which allow us to benefit from the availability of such efficient cryptographic operations as we discuss later with our proposal. Figure 1 also illustrates the reason why header compression is so prevalent in 6LoWPAN, because even when not using link-layer security applications do not have that much space left to transmit data.

3.2. Compressed security headers for 6LoWPAN

Because IEEE 802.15.4 does not provide any type of multiplexing information to allow a receiver to distinguish among different types of data packets, 6LoWPAN uses the first byte of the link-layer payload as a dispatch byte, which allows the identification of the transported packet and (if necessary) further information within the subtype. We need therefore to decide how new headers for security are going to be identified in 6LoWPAN using the dispatch byte. Three strategies would allow us to identify the presence of new headers in the context of 6LoWPAN, as we proceed to discuss.

The first option is to use the ESC header type value [8], which allows the usage of an additional dispatch byte to identify the presence of new headers. Using this approach the first (original) dispatch byte remains untouched and the following (new) dispatch byte can be used to identify new security headers. This approach presents the inconvenience of requiring one additional byte for this purpose. A second option is to use context-based header compression as in [17], particularly using the LOWPAN_IPCH and LOWPAN_NHC headers, and to define appropriate identification values for security using the IPv6 Extension Header Identifier (EID) field of the LOWPAN_NHC header. This approach is now viable because context-based header compression has been recently adopted as standard [21]. The third option is to integrate security in the context of standardized headers and identification values, by defining new dispatch type values for security using reserved values of the original payload byte. This is our approach and corresponds to a strategy identified from the start in RFC 4944 [8]. The usage of reserved dispatch values is both accepted and encouraged in this document, which defends that with the further development of 6LoWPAN additional functions

DOI: 10.1002/dac

Table II. New dispatch values to identify 6lowpan security and usage modes.

Header dispatch values for 6LoWPAN security	6LoWPAN security header and usage mode
01 001xxx	AH in transport mode
01 101xxx	AH in tunnel mode
01 011xxx	ESP in transport mode
01 100xxx	ESP tunnel mode

are expected to occupy unused space. We proceed by describing how such identification values are defined.

3.2.1. New 6LoWPAN dispatch type values for security. The IPv6 over Low Power Personal Area Networks uses the first two bits of the dispatch byte (the first byte of the IEEE 802.15.4 payload) to allow nodes to identify the presence of a 6LoWPAN packet or of other types of packets. For a 6LoWPAN packet, the remaining bits of the dispatch byte allow the identification of specific types of 6LoWPAN headers that correspond to given functionalities of the adaptation layer, namely a mesh, fragmentation or addressing header. When the first two bits identify a 6LoWPAN addressing header (value '01', please refer to Table II), several dispatch values are reserved as RFC 4944 [8] describes. We use four values from the set of reserved values to identify the presence of new 6LoWPAN compressed security headers and respective usage modes, as Table II describes.

The values in Table II are more precisely obtained from the set of reserved values after LOWPAN_HC1, which is the value defined to identify the presence of an HC1 compressed addressing header. HC1 is the header compression format adopted in 6LoWPAN to compress addressing information, while HC2 was defined to allow the compression of transport-layer User Datagram Protocol (UDP) header information. As we can see in Table II, the first three of the remaining 6 bits of the dispatch byte are sufficient to identify a security header, together with its usage mode and irrespective of the value of the remaining 3 bits. The three remaining bits are sufficient to distinguish between different types of 6LoWPAN addressing headers. This identification strategy therefore gives us the possibility of simultaneously identifying the presence of security and addressing information on a given 6LoWPAN packet, allowing also to save payload space and easing the processing of headers in tunnel and transport modes.

3.2.2. Compressed Encapsulating Security Payload header for 6LoWPAN. The design of new security headers for the 6LoWPAN adaptation layer must take into consideration several aspects. The first is that the principles of simplification, compression, and shared context around which other 6LoWPAN headers [8] were designed should also be considered for security. At the same time, it is desirable that the processing of such headers can be easily integrated into existing implementations of the IP Security architecture [22], because this would contribute to its evolution towards easily adopting new IPv6-enabled sensing applications. Another important aspect is that most sensing platforms currently possess or will probably adopt in the future hardware cryptographic operations. Hardware encryption and authentication must therefore be considered together with cryptographic algorithms implemented in software. For example, IEEE 802.15.4 requires hardware cryptography and platforms such as the TelosB [23] mote to support hardware security with the Advanced Encryption Standard (AES) cryptographic algorithm in CCM* combined mode using the cc2420 chip. AES/CCM provides encryption and decryption in the CTR (Counter) mode and authentication and integrity in the cypher block chaining message authentication code (CBC-MAC) mode. The CCM* variation of AES/CCM additionally offers encryption-only and integrity-only capabilities, a characteristic that makes it well adapted to the independent support of authentication and encryption headers. Considering that AES/CCM is part of the set of future mandatory algorithms for the IP Security architecture, we realize the importance of its consideration during the design of security headers for 6LoWPAN.

DOI: 10.1002/dac

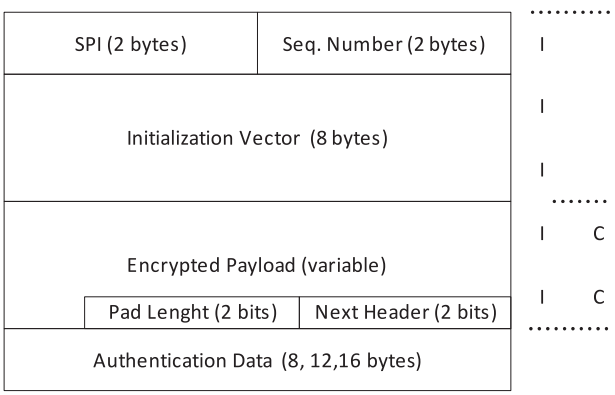

Figure 2. Compressed ESP security header for 6LoWPAN.

Because the design of new security headers for 6LoWPAN will require header compression, Internet hosts running a complete IPSec implementation are expected to support 6LoWPAN security headers in the future, and adapt to the usage of the compressed security fields in communications with constrained sensing devices. In Figure 2 we illustrate how the 6LoWPAN ESP [24] security header is built, and in the same figure we also illustrate which fields are integrity protected (with an 'I') and encrypted (or confidentiality) protected (with a 'C'). The purpose of this header is to provide applications with encryption and optional authentication and integrity of 6LoWPAN packets.

By analyzing the 6LoWPAN ESP header illustrated in Figure 2, we begin by identifying a 2-byte SPI (Security Parameters Index) field, whose purpose is to allow a receiving entity to relate an incoming packet to a specific security association. This allows a sensing device to obtain information such as the cryptographic algorithms and keys required to apply security operations to the packet. Given the constraints of sensing devices, a 2-byte SPI is considered appropriate. The next field stores a 2-byte sequence number, with the purpose of helping end systems in protecting against packet replay attacks. The sequence number is treated as an unsigned value and implementations must ensure that a distinct value is maintained for each different security association. Given the transmission rates of typical sensing applications, 2 bytes is considered appropriate for this field. Nevertheless, if necessary an option similar to ESN (extended sequence numbers) [24] may be designed for 6LoWPAN, allowing communicating parties to agree on larger sequence numbers. A 6LoWPAN ESN option would allow a device to maintain a larger sequence number for security associations requiring it, with such number being used for ICV-computation purposes, while only its lower 2 bytes being transmitted with each 6LoWPAN packet.

It is important to note in this context that other than the usage of a distinct sequence number for each security association, a key management mechanism appropriate to 6LoWPAN must be designed to allow keys to be periodically refreshed. This is due to the fact that algorithms such as AES/CCM completely lose its security if a given key is reused with the same sequence number. The development of appropriate key management mechanisms for 6LoWPAN appears therefore as an important research goal. For this purpose, Internet Key Exchange (IKE) can be simplified or in alternative completely different approaches to key management can be followed [25, 26].

After the ESP header we encounter the IV (initialization vector) field, which is used to transport cryptographic synchronization data necessary for two devices to successfully apply the same cryptographic algorithm. An 8-byte IV enables the usage of all the current and future mandatory cryptographic algorithms defined for the IP Security architecture. It also reflects recommendations from RFC 4309 [27], in that it allows compatibility with current and future cryptographic suites based on AES. Synchronization data are used as input to 3DES and AES in CBC mode algorithms or together with additional data generated by end devices to produce the input required for algorithms such as AES in CTR mode. CTR mode is available with hardware implementations of AES/CCM, and the rules currently defined for the usage of AES in CTR mode with

DOI: 10.1002/dac

the ESP header [27] state that 3 bytes of salt must be added to the IV data for this purpose, because AES requires an 11-byte nonce. Again, by following such rules we allow an easier integration of our new 6LoWPAN security headers in current implementations of the IP Security architecture.

Next in the packet comes the encrypted data, at the end of which two fields are added that aid in employing the security header with different encryption algorithms and usage modes. The first is the pad length field, which stores the number of padding bytes (from 1 to a maximum of 4) added to the original encrypted data to align up the payload and trailer, if required by the encryption algorithm employed. Next appears the next header field, which stores information on how the receiver should interpret the decrypted data by indicating the presence of a Transmission Control Protocol (TCP), UDP, or Internet Control Message Protocol version 6 (ICMPv6) packet.

At the end of the 6LoWPAN ESP header follows the ICV (integrity check value) or MIC (message integrity code) field, which stores the authentication data used to authenticate the origin of the 6LoWPAN packet and verify its integrity. Because such operations are optional with the ESP header they are only performed if required in the context of a given security association. The size of the authentication data depends on the encryption algorithm used to generate the MIC code and on the level of integrity and authentication required for the given security association. This field is of 12 bytes if generated using HMACSHA196 or AES-XCBC-MAC-96, because both algorithms produce a 96-bit MIC code. When using hardware AES/CCM, this algorithm can be used to generate 8, 12, or 16 bytes MIC codes, also in line with recommendations from RFC 4309 [27]. The MIC code is not protected by encryption, meaning that smart objects are able to verify the authenticity and integrity of a received 6LoWPAN packet protected with ESP before being required to perform more computationally demanding decryption operations.

The layout of the 6LoWPAN ESP header reflects the design principles adopted by 6LoWPAN. Simplification and compression are performed whenever possible, while at the same time the relevant fields are appropriately dimensioned for software and hardware cryptography.

3.2.3. Compressed authentication header for 6LoWPAN. The purpose of the 6LoWPAN authentication header (AH) is to allow end systems that do not require confidentiality to verify the integrity and origin of a given 6LoWPAN network-layer packet, and also to provide protection against replay attacks. The usage of the authentication header is of interest because such operations are less demanding of resource-constrained smart objects than encryption and decryption. Many sensing applications on the IoT will probably not require encryption, because the data transported is itself not confidential, while the most important will be to protect communications against corrupted packets and to authenticate its origin. In Figure 3 we illustrate the 6LoWPAN compressed AH [28], and in the same figure we also indicate which parts of the header and data payload are integrity and authentication protected (as indicated by an 'A') and are considered mutable (as indicated by an 'M') or immutable fields (as indicated by an 'I') in respect to the computation of the ICV. Mutable fields are fields for which the sending device (which must compute the ICV) is unable to calculate or

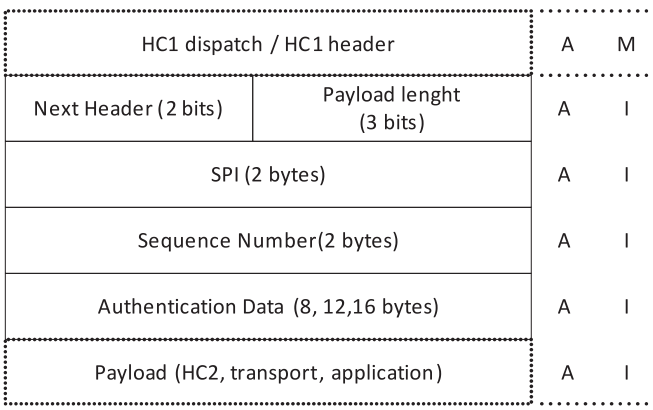

Figure 3. Compressed AH security header for 6LoWPAN.

DOI: 10.1002/dac

predict its final value upon arrival of the packet at its destination, and thus such values are zeroed for IVC computation purposes. An added advantage of the 6LoWPAN authentication header over its ESP counterpart is that security (authentication and integrity) can also be applied to fields outside the security header itself. The authentication data are computed considering all the fields identified as immutable in Figure 3 (with the ICV field itself being zeroed for that purpose), but implementations may also decide to include immutable fields of the HC1 or HC2 addressing and transport headers. The padding required by the integrity algorithm (if any) and the high-order bits of the ESN option (if adopted for 6LoWPAN in the future, as previously discussed) are also considered during the computation of the ICV.

Analyzing the 6LoWPAN AH header illustrated in Figure 3, the next header field allows the identification of the next header as TCP, UDP, or ICMPv6. Similarly to the original authentication header [28], the payload length field stores the total length of the header in units of 32-bit words. As Figure 3 illustrates, 3 bits are sufficient to measure the space necessary to store the authentication data (maximum 16 bytes), sequence number, SPI, next header, and payload length fields. To achieve byte alignment, implementations should consider that the next header and the payload length fields occupy one byte and zero the remaining bits. Byte alignment of the authentication header promotes higher efficiency in header processing by 6LoWPAN implementations. We followed this rule in the TinyOS implementation evaluated later in the paper. The SPI field again allows a device to map the 6LoWPAN packet to a particular security association, and the sequence number supports protection against packet replay attacks. The size of the authentication field is proportional to the integrity and authentication level required for the security association and is in line with the set of cryptographic algorithms that can be used for its generation, as was previously discussed for the ESP header and utilized in our experimental evaluation study later in the paper. As a final remark concerning the 6LoWPAN security headers described, one aspect to note is that they do not allow the maintenance of nice 32-bit or 64-bit boundaries that were a concern during the design of its counterparts [24, 28] for the IP Security architecture. This is not so much of a problem for 6LoWPAN because it is designed to be implemented in sensing platforms typically employing low-end 8-bit or 16-bit microcontrollers.

3.2.4. Usage of security in the context of existing 6LoWPAN headers. Because other 6LoWPAN headers are currently defined, we need to consider how the new 6LoWPAN security headers are allowed to be employed side-by-side with them. We need therefore to analyze the usage of security together with the mesh addressing header, the fragmentation header and the compressed addressing header. This is important not only in respect to the implementation of a security-enabled 6LoWPAN networking stack, but also because it allows us to investigate the impact of security on the final payload space available to 6LoWPAN applications, as we do in our evaluation study later in the paper.

The mesh-addressing header transports information for layer-two forwarding whenever a mesh-routing protocol is employed for routing packets from node to node in the LoWPAN [29]. It is important to note that mesh-routing is independent of 6LoWPAN, because IPv6 only cares about the source and destination addresses of the devices, independently of how the packet arrives at its destination. The fragmentation header transports information related to how the original IPv6 packet was fragmented for its transportation in the 6LoWPAN, and which therefore is necessary for the reassembly of the original packet at the destination node. Finally, the compressed addressing header allows the compression of IPv6 addresses and multicast addresses whenever possible.

Considering that 6LoWPAN security is inherently end-to-end, meaning that it is intended to be generated and interpreted by 6LoWPAN devices, the headers that are destined to be interpreted by each device on the path of the 6LoWPAN packet towards its final destination must not be considered for security purposes. This applies to the mesh addressing header, which therefore must appear before any 6LoWPAN security header independently of its usage mode. The same rationale can be applied to a broadcast (LOWPAN_BC0) and fragmentation headers. A broadcast packet stores a sequence number intended to be interpreted at each forwarding node, allowing the implementation of the broadcast mechanism using a flooding communications algorithm. The fragmentation header transports information necessary for the reassembly of the IPv6 packet at the

DOI: 10.1002/dac

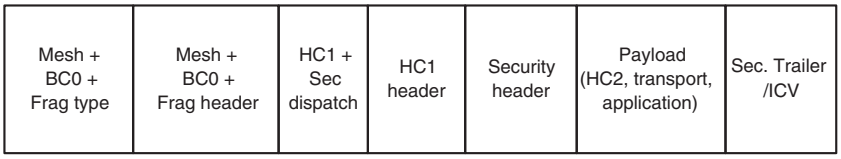

Figure 4. Usage of 6LoWPAN security in transport mode.

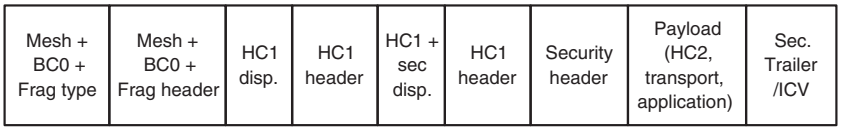

Figure 5. Usage of 6LoWPAN security in tunnel mode.

6LoWPAN destination. In summary, we consider that 6LoWPAN security headers protect only end-to-end payloads as makes sense for network-layer security, and as such appear after the mesh, broadcast, and fragmentation headers.

As with the traditional IP Security architecture, we consider that 6LoWPAN security may be useful in two different usage modes: the tunnel mode and the transport mode. Transport mode enables secure communications between two end devices (smart object or other type of 6LoWPAN or IPv6 device) and will be preferred in many usage scenarios, also considering that it requires less header space from the (already limited) link-layer payload. On the other end, tunnel mode allows for the tunneling of secure communications via intermediate devices functioning as security gateways or as 6LoWPAN routers. The usage of 6LoWPAN security in these two usage modes is discussed in greater detail next.

3.2.5. Tunnel and transport mode usage scenarios. In Figure 4 we illustrate the usage of 6LoWPAN security in transport mode, side-by-side with other compressed 6LoWPAN headers and data from transport protocols and applications. As previously discussed, the mesh, broadcast, and fragmentation headers (when present) appear before the security headers. Security is identified side-by-side with compressed addressing, and in transport mode acts on the payload of the original packet, which may contain an HC2 compressed UDP header and data from other transport protocols and applications. The scope of this 6LoWPAN security header in transport mode depends on it being AH or ESP, as previously discussed. When using authentication and integrity, a MIC or ICV is transmitted at the end.

Regarding the usage of 6LoWPAN security in tunnel mode, two addressing headers are necessary and security is employed as we illustrate in Figure 5. The inner addressing header identifies the address of the ultimate destination of the 6LoWPAN packet, which may be for example a 6LoWPAN smart object, while the outer addressing header identifies the immediate (intermediate) destination of the packet, for example a security gateway placed between the Internet and the network of smart objects supporting a given sensing application, or a 6LoWPAN router supporting secure communications between remotely deployed sensing devices.

The usage of 6LoWPAN security in tunnel mode allows the protection of the entire inner (ultimate) addressing header and also of the original data payload. Again, which fields are considered for security depends on the usage of AH or ESP, and may also depend on particular implementations of 6LoWPAN security, because networking stacks may decide to include information from compressed transport headers such as HC2. As with security in transport mode, authentication data follow at the end if necessary.

4. EXPERIMENTAL EVALUATION SETUP

The validation of any proposal on security for resource-constrained sensing devices is of particular relevance if performed experimentally, because in practice several unpredicted aspects related to the

DOI: 10.1002/dac

functioning of sensing devices and wireless communications are difficult to reproduce realistically using simulation environments. Such aspects motivated the experimental evaluation of our proposal. We proceed by describing the experimental evaluation scenario and discussing conclusions obtained from our experimental measurements.

4.1. Experimental evaluation scenario

Our proposal on security for the 6LoWPAN adaptation layer was implemented using the TinyOS operating system [5, 30], more precisely by modifying its Berkeley Low-power IP (BLIP) [31] networking stack. For the experimental measurement of the values relevant for our evaluation study, we employed UDP communication sessions established between different 6LoWPAN devices, in particular between a TelosB [23] mote and a Linux host supporting both 6LoWPAN and IPv6. The Linux host is a router between an Ethernet IPv6 network and the IEEE 802.15.4 LoWPAN, employing a second TelosB mote as a bridge supporting communications with the network of smart objects. This Linux 6LoWPAN router also supports routing advertisements for the 6LoWPAN. The TelosB mote used for the experimental evaluation of the measured parameters is a battery-powered device supporting our TinyOS testing application and the 6LoWPAN security-capable BLIP networking stack. The TelosB mote is currently a popular research platform providing a good reference for the validation of our proposal, because it is a good representative of the computational power currently available with commercial sensing platforms. In particular, the TelosB is powered by a 16-bit RISC MSP 430 microcontroller with 10 kB of random-access memory (RAM) for program execution and 48 kB of read-only memory (ROM) for program storage. It also supports communications at 2.4 GHz and data transmissions at 250 kb/s. Because it implements the IEEE 802.15.4 standard, it also provides hardware encryption and authentication using the AES/CCM cryptographic suite, which we consider in our proposal and employ during our evaluation study.

4.2. Identification of appropriate cryptographic algorithms

The selection of the cryptographic algorithms that are appropriate to support 6LoWPAN security and to the resources of smart objects is an important requirement for our experimental study. Such algorithms or suites of algorithms enable smart objects to perform encryption, decryption and integrity/authentication related operations, therefore allowing the processing and generation of information transported with 6LoWPAN using security headers. For the identification of the appropriate cryptographic suites our goal is in fact twofold, because on the one side the usage of algorithms that are already accepted in the IP Security architecture would facilitate the integration of sensing applications with the Internet in a secure fashion, while on the other side we must carefully consider the effectiveness of the usage of such algorithms in resource-constrained smart objects. The selection of cryptographic algorithms regarding its impact on smart objects must nevertheless not be too conservative in this respect, because it may be expected that sensing devices will become more powerful and energy-efficient in the near future [7], and thus security mechanisms that have been shown to be unviable or marginally viable in the present may well be employed in a more mainstream fashion using future sensing platforms.

Because the current IP Security architecture [22] may evolve to include 6LoWPAN applications in the future, we find it useful to analyze the effectiveness of the usage of its mandatory cryptographic algorithms with smart objects. The same applies to the algorithms that will most probably be adopted as mandatory in the future. The fact that the IP Security architecture allows end systems to agree on security algorithms and related security configuration parameters at the establishment of a security association is also in line with our requirement of adaptability for 6LoWPAN security. Adaptable security mechanisms at the network layer may allow a 6LoWPAN smart object to select a cryptographic algorithm from a pool of alternatives and to decide how to use that algorithm, and this serves our goal on providing security mechanisms that allow the establishment of acceptable compromises between security and resources required from smart objects, two aspects we consider important for the IoT. In Table III we identify the cryptographic algorithms that are either currently defined as mandatory for the IP Security architecture [22] or that will probably be adopted as such in the near future [32].

DOI: 10.1002/dac

Table III. Current and future mandatory cryptographic algorithms for the IP Security architecture.

Security header	Cryptographic algorithm	Usage	Status
ESP	3DES-CBC	Encryption	Mandatory
	AES-CBC	Encryption	Future
	HMAC-SHA1-96	Authentication	Mandatory
	AES-XCBC-MAC-96	Authentication	Future
	AES-CCM	Combined	Future
AH	HMAC-SHA1-96	Authentication	Mandatory
	AES-XCBC-MAC-96	Authentication	Future

Table IV. Usage scenarios of cryptographic algorithms and 6LoWPAN security headers.

Cryptographic suites	6LoWPAN header	Security provided
3DES-CBC AES-XCBC-MAC-96	ESP	Confidentiality Integrity, authentication
3DES-CBC HMAC-SHA1-96	ESP	Confidentiality Integrity, authentication
AES-CBC AES-XCBC-MAC-96	ESP	Confidentiality Integrity, authentication
AES-CBC HMAC-SHA1-96	ESP	Confidentiality Integrity, authentication
AES/CCM (HW)	ESP	Confidentiality, integrity, authentication
AES-XCBC-MAC-96	AH	Integrity, authentication
HMAC-SHA1-96	AH	Integrity, authentication
AES/CCM (HW)	AH	Integrity, authentication

As we can see in Table III, a shift is expected to take place towards AES-based cryptographic solutions. This is also in line with the fact that AES is already supported by various sensing platforms, and also motivated our decision on considering the usage of AES/CCM during the design of the 6LoWPAN security headers. The AES CCM* mode available with sensing platforms such as the TelosB allows for the separate support of security operations as required for the suites that employ AES in Table III.

Because the usage of standard security and communications mechanisms may facilitate the secure integration of sensing applications with the Internet, our experimental evaluation study considers the usage of the algorithms in Table III in obtaining network-layer 6LoWPAN security. In Table IV we describe how the above algorithms are employed in support of security using the compressed ESP and AH 6LoWPAN headers.

As Table IV reflects, the isolated testing of the algorithm described in Table III would not be appropriate to evaluate the effectiveness of 6LoWPAN security, because in most deployments at least two algorithms will need to be supported, one providing confidentiality (throughout encryption and decryption) and the other providing authentication and integrity (throughout generation and verification of a MIC code or secure hash). AES/CCM was tested as available at the hardware in the TelosB mote, while the other algorithms were programmed in software using code optimized for small microcontrollers with the characteristics of the MSP 430.

The cryptographic block size and key size used with each algorithm are the values inherent of each cryptographic algorithm itself, and are also in line with the configurations required by the IP Security architecture. Such values constitute therefore the most appropriate configuration to measure the effectiveness of our proposal. In particular, 3DES-CBC uses 128-bit keys to process 64-bit blocks. AESCBC, AESXCBCMAC96, and AES/CCM (in hardware) use 128-bit keys to process 128-bit blocks. HMACSHA196 uses 160-bit keys to process 512-bit blocks, with the

DOI: 10.1002/dac

original 160-bit authenticator generated being truncated to 96 bits, as specified in RFC 2404 [33]. Our AESCBC software implementation also supports the AESXCBC-MAC-96 algorithm, with the XCBC mode modifying the classic CBC mode as documented in RFC 3566 [34].

The fact that our tests employ software-based and hardware-based cryptographic algorithms allows us to analyze the feasibility of 6LoWPAN security for a broader set of devices. This is relevant also if we consider that the IoT will include heterogeneous sensing devices, which may or may not support hardware security. In the evaluation study we describe next we consider the usage of ESP to provide confidentiality together with authentication and integrity. Although we could have considered using ESP only for confidentiality, we believe that authentication and integrity are security properties that will be required for most of the applications in the IoT. In fact, the opposite may be truer, in that many applications will probably be able to dispense confidentiality and use only AH with its authentication and integrity assurances.

5. EXPERIMENTAL EVALUATION OF 6LOWPAN SECURITY

Our evaluation on the feasibility of 6LoWPAN security begins by analyzing its impact on the 6LoWPAN payload space. Later in the paper we concentrate on aspects such as its energy and computational requirements, which are determinant for the achievement of acceptable transmission rates and lifetimes for sensing applications.

5.1. Overhead of security on 6LoWPAN payload space

Because the payload space available to applications is an important factor in dictating the usefulness of 6LoWPAN in real usage scenarios, we start by analyzing the packet overhead of the usage of security in both tunnel and transport modes. We start by analyzing the payload space required for 6LoWPAN in various addressing compression scenarios and also with mesh and fragmentation headers. We must also consider the payload space required for the security headers previously described. The space required for such 6LoWPAN headers is described in Table V. The values illustrated in this Table are used during our following analysis on the impact of security on 6LoWPAN payload space.

The first three lines of Table V refer to the possible address compression scenarios that 6LoWPAN allows. With link-local unicast communications between 6LoWPAN smart objects sharing the same local link address, HC1 and HC2 6LoWPAN compression allows the compression of an UDP/IPv6 header down to 7 bytes. In this scenario the version, traffic class, flow label, payload length and next header fields, and also the link-local prefixes of the IPv6 source and destination addresses are all elided, with the correspondent IPv6 suffixes being derived from the IEEE 802.15.4 header. The second compression scenario corresponds to communications with an object outside of the local link while on the same 6LoWPAN, and in this case the IID (Interface Identifier) suffix of the source and destination addresses is also obtained from IEEE 802.15.4 addressing information, but the source and the destination prefixes must be carried inline. At the end, an additional 16 bytes are required for addressing information. The third scenario is also the most useful in the context of the IoT, as in this case a 6LoWPAN-enabled smart object is able to communicate directly with an Internet host

Table V. Payload space requirements for 6LoWPAN addressing, mesh, fragmentation and security.

Scenario	Payload requirement
Link-local unicast	7 bytes
Outside of link-local scope	23 bytes
Outside of local LoWPAN	31 bytes
6LoWPAN AH	37 bits
6LoWPAN ESP	96 bits
Fragmentation	4 bytes / 5 bytes
Mesh addressing	5 bytes / 17 bytes

DOI: 10.1002/dac

or with another remote smart object. In this scenario, 6LoWPAN is only able to elide the source address IID, with the remaining part of the source address and with the full destination IPv6 address carried inline, requiring in total 31 bytes.

The remaining lines in Table V refer to the payload space required for the other 6LoWPAN headers, including the two new security headers illustrated in Figures 2 and 3. Without considering the transportation of encrypted and authentication data, the authentication header requires 37 bits and the ESP header requires 96 bits. Fragmentation requires 4 bytes for the first fragment and 5 bytes for subsequent fragments, while the mesh addressing header required 5 or 17 bytes, depending on the usage of short (16-bit) or long (EUI-64 64-bit) addresses, respectively. Such values are also represented in Table V and are considered for the following analysis on the impact of 6LoWPAN security on packet payload space.

5.1.1. Impact of 6LoWPAN security without fragmentation and mesh headers. We illustrate in Figure 6 the impact on the 6LoWPAN payload space of security in tunnel and transport modes, without considering the usage of fragmentation or mesh addressing headers. We consider the addressing compression scenarios previously discussed and the transportation of authentication data of 8, 12, and 16 bytes in length. Figure 6 illustrates the payload space available in percentage of the maximum of 102 bytes available with IEEE 802.15.4 without link-layer security. We also illustrate the payload space available when using 6LoWPAN headers without any security-related header or data. Because fragmentation is not considered, the values illustrated in Figure 6 correspond to the maximum payload space that applications not using mesh addressing can use without requiring fragmentation from the 6LoWPAN adaptation layer.

Because we considered the values from Table V for each of the three addressing compression scenarios, the payload space required for HC1 and HC2 compressed headers is already accounted for. It is visible that transport mode security is clearly less expensive than tunnel mode security in terms of the payload space required. For link-local communications or communications with systems outside of the local link but on the same 6LoWPAN, security leaves from 51 to 82 bytes to 6LoWPAN applications using ESP or AH in transport mode. When communications with the outside of the 6LoWPAN are required, the available space also in transport mode is between 43and 58 bytes. Security in transport mode therefore provides acceptable availability on payload space, regardless of the security header and of the integrity and authentication level. Considering that tunnel mode in reality is not useful for link-local communications, we see that for communications outside of the local link the payload available is between 28 and 43 bytes and for communications outside of the 6LoWPAN it is between 12 and 27 bytes. Therefore, tunnel mode security for communications between devices on different LoWPANs is viable mainly for applications requiring moderate

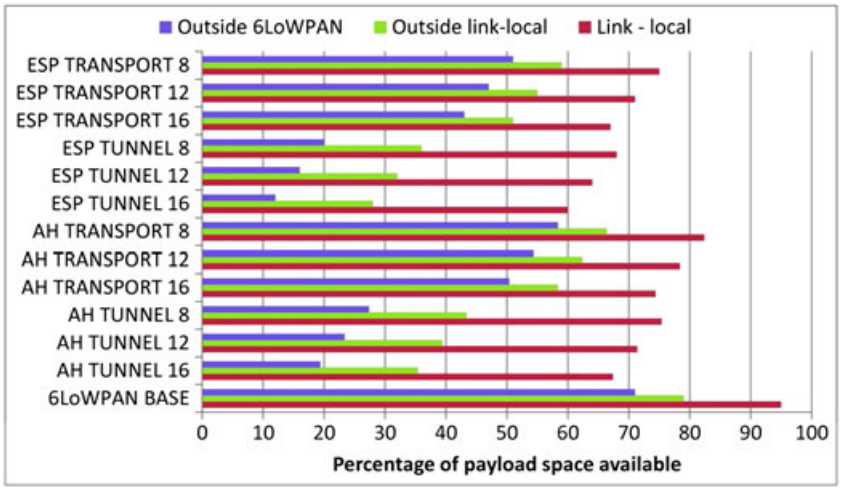

Figure 6. Payload space available with 6LoWPAN security without mesh or fragmentation headers.

DOI: 10.1002/dac

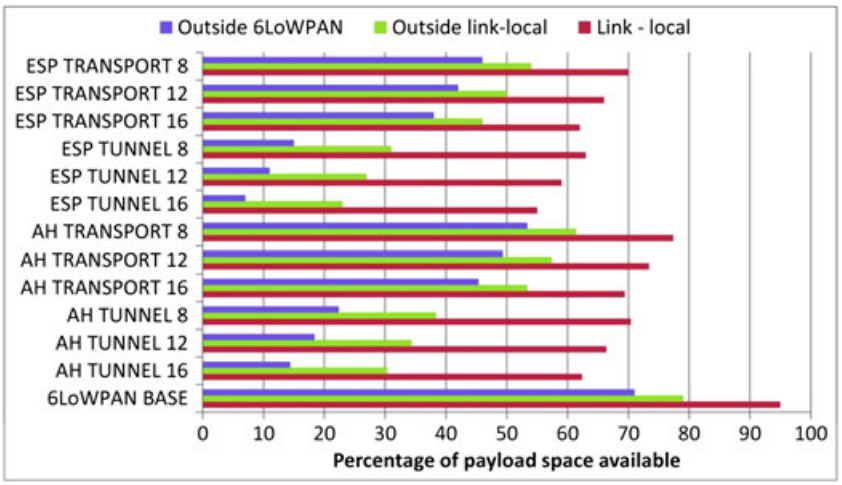

Figure 7. Payload space available with 6LoWPAN security with fragmentation information.

amounts of data. For communications with devices outside of the 6LoWPAN tunnel mode is viable but only for applications requiring the transportation of only a few bytes.

5.1.2. Impact of 6LoWPAN security with fragmentation. Our next evaluation considers the usage of a fragmentation header, and the obtained values are illustrated in Figure 7. Because the overhead imposed from the fragmentation header is only of 5 bytes per 6LoWPAN packet, our previous conclusions remain valid regarding security in transport mode, as the payload space remains between 38 and 77 bytes.

As for tunnel mode security, it leaves between 23 and 38 bytes for communications with nodes outside the local link, and between 7 and 22 bytes for communications with other 6LoWPAN or IPv6 hosts. We are therefore able to realize that for tunnel mode the space required for the fragmentation header poses an extra pressure on the usefulness of this security mode. ESP in tunnel mode can only be considered viable if employed with an 8-byte or 12-byte MIC code.

In conclusion, communications requiring fragmentation can use transport mode security viably with all addressing compression scenarios. Tunnel mode security is valid for communications with nodes outside of the local link for applications requiring the transmission of small amounts of sensing data, while for communications with Internet hosts it is viable mainly for applications that do not require confidentiality and therefore are able to use AH to protect the transportation of small amounts of data. For applications that do require confidentiality, ESP is a viable choice only if lower integrity and authentication assurances are acceptable, more precisely using ESP with a MIC code with 12 or (preferably) 8 bytes.

5.1.3. Impact of 6LoWPAN security with mesh addressing. Figure 8 illustrates the impact of 6LoWPAN security on payload space when the transportation of mesh addressing information is required. We consider the usage of a mesh-addressing header with 17 bytes, corresponding to mesh addresses obtained from the EUI-64 addresses of sensing devices.

We can observe that with mesh addressing and security in transport mode the payload space available for 6LoWPAN applications drops to between 26 and 65 bytes. As for tunnel mode security, communications with nodes outside of the local link remain possible if small amounts of data are required to be transmitted, as in this case only from 11 to 26 bytes are available. For communications with nodes outside of the 6LoWPAN, tunnel mode is viable only with AH transporting MIC codes with 8 bytes, which even so only provides 10 bytes of payload space. The remaining tunnel security usage modes do not provide enough payload space, or the support of 6LoWPAN security headers would require the availability of more than 102 bytes. In conclusion, in the presence of mesh addressing information, security in transport mode remains valid but only for applications

DOI: 10.1002/dac

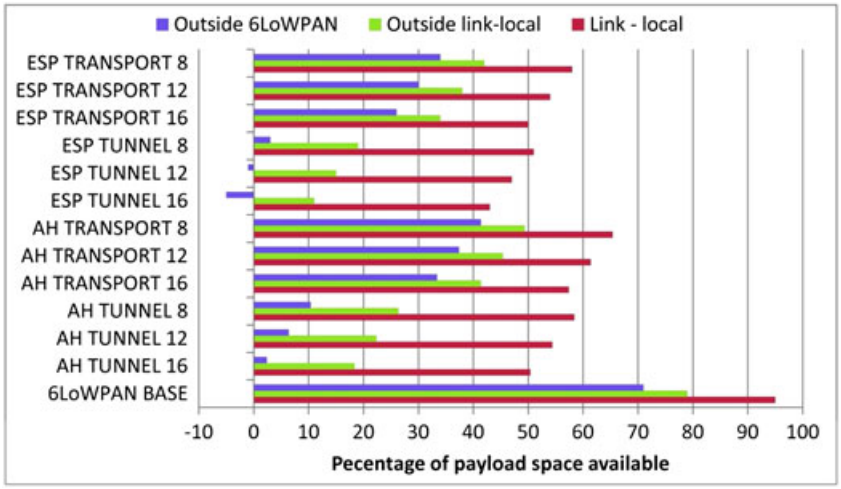

Figure 8. Payload space available with 6LoWPAN security with mesh information.

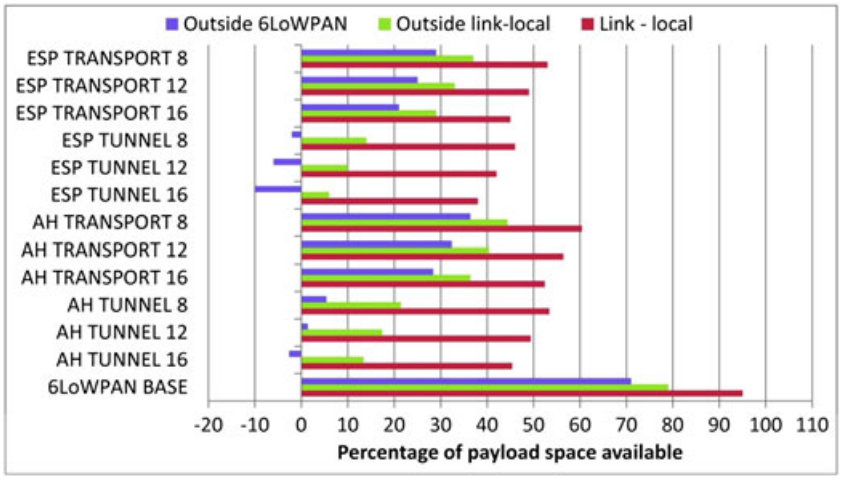

Figure 9. Payload space available with 6LoWPAN security with fragmentation and mesh information.

requiring the transmission of a moderate amount of data. On the other end, tunnel mode security is viable only for applications requiring low integrity and authentication assurances or which do not need confidentiality at all.

5.1.4. Impact of 6LoWPAN security with fragmentation and mesh information. The worst usage scenario for 6LoWPAN security in terms of its impact on payload space corresponds to the simultaneous usage of fragmentation and mesh addressing headers, and is illustrated in Figure 9. The values illustrated in Figure 9 corroborate some of our previous conclusions. We are able to conclude that transport mode security remains a valid usage mode for small amounts of data, as in this case between 21 and 60 bytes are available to transport data from 6LoWPAN applications. Tunnel mode is clearly the most affected mode by the lack of available payload space, and in practice can be considered unviable for communications with nodes outside of the 6LoWPAN, because in this case even AH with a MIC code of 8 bytes would only leave 5 bytes of data payload space, which may be insufficient for most sensing applications on the IoT. We can see that with several configurations there is not enough space to accommodate even just the 6LoWPAN headers.

Regarding tunnel mode communications with nodes outside of the local link, it still can be considered viable for very small amounts of transmitted data, because between 6 and 21 bytes

DOI: 10.1002/dac

Table VI. Viable usage scenarios for 6LoWPAN security in the context of the IoT.

	To		
From	6LoWPAN device on same local link	6LoWPAN device outside of local link	Device outside the 6LoWPAN
6LoWPAN device	AH/ESP in transport mode	AH/ESP in transport mode	AH/ESP in transport mode
		AH/ESP in tunnel mode via 6LoWPAN router	AH/ESP in tunnel mode via security gateway

are available. This is especially true for applications that only require authentication and integrity, because with AH in tunnel mode between 13 and 21 bytes are available.

5.1.5. Viable usage modes of 6LoWPAN security. Other than the identification of the viable usage modes with respect to the impact of security on 6LoWPAN payload space, we need to identify the usage modes of security that will in fact be useful in the context of the IoT. Table VI identifies such modes, from the perspective of communications initiated by a 6LoWPAN-enabled smart object.

The scenarios identified in Table VI consider the usage of two types of 6LoWPAN routers, one acting as a 6LoWPAN security gateway and the other as a 6LoWPAN router. A security gateway is a device without the resource constraints that are typical of smart objects, and that as such can be used to aid in the integration and interconnection of a network of smart objects with the Internet. A security gateway may implement various security mechanisms to protect the network of smart objects from the Internet, among which the processing of network-layer security in communications with Internet devices and smart objects. On the other end a 6LoWPAN router is a more limited device, supporting distributed sensing applications and allowing routing and enforcing security mechanisms for communications between different 6LoWPANs. Our previous evaluation and the identification of the useful usage modes of 6LoWPAN security allow us to identify the main characteristics of the viable usage modes of the proposed security headers. In this context, viability means that enough payload space is left for applications while guaranteeing the usage of strong authentication codes. It is clear that, without employing a mesh routing protocol, 6LoWPAN network-layer security is viable in all usage modes as long as applications are able to adapt to the payload space available.

The classification in Table VII reflects a qualitative evaluation for which preference is given to the usage of strong authentication and integrity codes whenever possible. Other practical usage scenarios can nevertheless be identified to be viable for the IoT, for example considering that some applications may only need to use smaller authentication codes or use ESP without authentication and integrity.

5.2. Memory footprint of 6LoWPAN security

Because memory is also a limited resource on smart objects, our evaluation study proceeds with the analysis of the memory footprint of our implementation of 6LoWPAN security in TinyOS and BLIP, while supporting the cryptographic suites previously identified. Different versions of a base TinyOS application were employed in our experimental evaluation, supporting the security-enabled 6LoWPAN stack together with each of the cryptographic suites implemented in software or available in hardware. We separately measured the RAM and ROM necessary with each version of the testing application, because both types of memory are limited on sensing devices.

In Figure 10 we describe the memory footprint of 6LoWPAN security with each of the cryptographic suites. The values illustrated in this figure are in percentage of the total of RAM and ROM

DOI: 10.1002/dac

Table VII. Viable usage modes of 6LoWPAN network-layer security.

							Available payload space			MIC code size		
	Mesh	Frag	Transport	Tunnel	AH	ESP	High	Medium	Low	16	12	8
Link-local			•		•	•	•			•		
		•	•		•	•	•			•		
	•		•		•	•		•		•		
Outside local link			•		•	•	•			•		
				•	•	•		•		•		
		•	•		•	•	•			•		
		•		•	•	•			•	•		
	•		•		•	•		•		•		
	•			•	•	•			•	•		
	•	•	•		•	•			•	•		
	•	•		•	•				•	•		
Outside 6LoWPAN			•		•	•	•			•		
				•	•	•			•	•		
		•	•		•	•	•			•		
		•		•	•				•	•		
		•		•		•			•		•	
	•		•		•	•		•		•		
	•			•	•				•			•

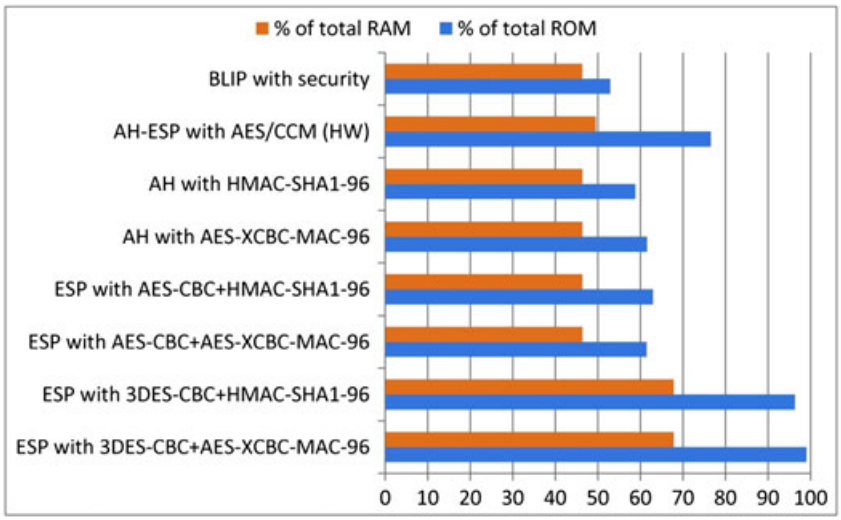

Figure 10. Memory footprint of 6LoWPAN security.

memory available on the TelosB (10 kB of RAM and 48 kB of ROM). For comparison purposes, we also evaluate and illustrate the memory required for a base BLIP networking stack with support for the processing of 6LoWPAN security headers but without any cryptographic algorithm. This base application allows us to measure the impact of the different cryptographic algorithms on the memory required from a sensing device.

When compared with the baseline usage profile, we can observe that ESP using cryptographic suites based on 3DESCBC, together with HMAC-SHA1-96 and AESXCBCMAC-96, is very demanding particularly in terms of the required ROM, leaving almost no ROM left available to accommodate other mechanisms or applications. The large ROM footprint of 3DES-CBC is mostly

DOI: 10.1002/dac

due to the usage of large S-Boxes by the algorithm. We can also note that the usage of the hardware-level encryption does not come without a non-negligible overhead on memory, particularly in terms of ROM, because code is necessary to support the usage of link-layer standalone encryption using the cc2420 chip of the TelosB. Our tests were performed using the standalone hardware encryption code available from the Shanghai Jiao Tong University [35]. In contrast to the inline mode, standalone encryption allows applications to perform hardware encryption and decryption without requiring the transmission or reception of a packet by the link-layer, given that such operations are controlled at a higher level in BLIP. From Figure 10 we can also observe that security suites based on the usage of AES-CBC with HMACSHA196 or AESXCBCMAC-96 broadly present a similar impact on the required ROM, while requiring only a few more bytes of RAM compared with the base 6LoWPAN security application.

From this analysis we are therefore able to conclude that, regarding requirements of memory available in resource-constrained sensing devices, AES appears as a natural candidate in providing an alternative to 3DES-based security suites. AES provides good security both in the CCM and CBC modes with a lower memory footprint. Of course, the usage of AES/CCM on devices that support hardware encryption presents the advantage of freeing more memory for other mechanisms and applications, and in this case AES-CBC can probably be dispensed. Regarding the support of integrity and authentication, AESXCBC-MAC-96 represents a good choice regarding the required memory, also because it provides superior security to HMAC-SHA1-96 with a similar memory footprint. It is interesting to note that, excluding the cryptographic suites using 3DES-CBC, security in general causes a relatively low overhead in terms of memory. The impact on the available memory of sensing devices therefore does not compromise the adoption of network-layer security mechanisms in the context of the 6LoWPAN adaptation layer.

5.3. Energy overhead of 6LoWPAN security

Because many sensing applications are designed with battery-powered sensing devices in mind, the energy required from such devices to perform security operations is a critical aspect, given that it influences the expected lifetime of the device and of the overall sensing application. Energy is therefore an important evaluation criterion of the feasibility of any communications or security proposal for smart objects, and one that we evaluate for the usage of 6LoWPAN security.

In Figure 11 we represent the experimentally obtained values of the energy required to process security for a 6LoWPAN packet with 32, 64, 96, or 102 bytes, using the previously identified

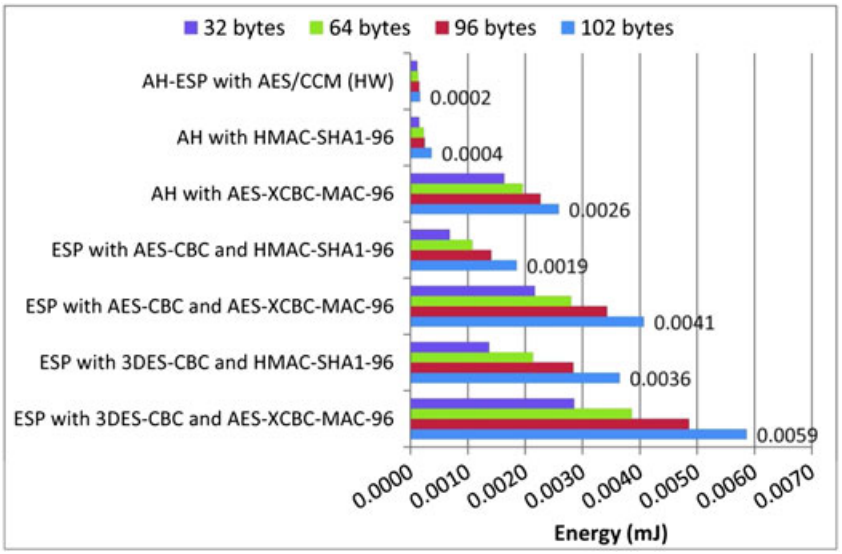

Figure 11. Energy required by 6LoWPAN security.

DOI: 10.1002/dac

cryptographic suites. The energy values represented are in millijoules (mJ), and the labels illustrate the energy required for the processing of security in the case of a fully sized 102 bytes 6LoWPAN packet. 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. Figure 11 allows us to perform a qualitative analysis of the impact of security on the energy available in smart objects, while the values obtained experimentally are later used in the context of our quantitative study on the lifetime of particular sensing applications. Please note that the values represented in Figure 11 already include the energy required for the processing of 6LoWPAN security headers (for its interpretation and construction) in the BLIP networking stack. Also, note that we do not represent the energy required for the processing of a 6LoWPAN packet without any cryptographic operation, because such value is negligible when compared with the energy required for security. The values represented are considered irrespective of the size of the MIC code generated by a specific authentication algorithm. This is due to the fact that AES-XCBC-MAC-96 and HMAC-SHA196 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 the same, as in this case hardware security is designed to operate on blocks of 128 bits (16 bytes).

From Figure 11 we again observe that cryptographic suites employing 3DES-CBC are clearly less efficient in terms of the energy required, as for example 0.0059 mJ are required to encrypt a 102-byte 6LoWPAN packet and generate the correspondent MIC code using AESXCBC-MAC-96. Regarding the support of authentication and integrity, the difference between HMAC-SHA1-96 and AESXCBCMAC96 is notorious, which allows us to conclude that the bigger security provided by AES-XCBC-MAC-96 probably does not compensate its impact on energy, when compared with the alternative HMAC-SHA1-96. In fact, HMAC-SHA1-96 only requires 0.00037 mJ to encrypt a 102-byte 6LoWPAN packet, while with AES-XCBC-MAC-96 0.0026 mJ are required to process the same packet. From Figure 11 we can also confirm that standalone hardware encryption using the cc2420 chip of the TelosB is extremely energy-efficient, and should therefore provide a superior solution to support integrity, authentication and encryption for 6LoWPAN security in devices where hardware security is available. As expected, encryption using AES/CCM is also clearly superior to AESCBC implemented in software. It is also interesting to note the superior performance of HMACSHA1-96 even when implemented in software, because in reality it is not much more expensive than hardware-based AES/CCM. Finally, our measurements reveal a better performance in terms of energy when compared with [17], although we must note that in such proposal energy is estimated and authors only consider link layer security and data payloads of up to 64 bytes. Our measurements are obtained experimentally and, because our overall goal is to evaluate the effectiveness of network-layer security in the context of the IP Security architecture, we also address the other (current and future) mandatory security suites.

5.4. Computational overhead of 6LoWPAN security

Other than the memory and energy required to process 6LoWPAN security, the computational effort required from smart objects for security operations is also a relevant aspect. Because advanced mechanisms such as multithreading are usually not supported in low-end microcontrollers such as the MSP430 of the TelosB, the computational time required to process security for a 6LoWPAN packet directly influences the maximum communications rate that a smart object can expect to achieve for a given sensing application.

In Figure 12 we illustrate the computational time required for the processing of a 6LoWPAN packet of different sizes, considering the cryptographic suites previously identified. The values illustrated are in milliseconds (ms) and, as with our previous analysis, we do not represent the computational time required for the processing of 6LoWPAN security without any cryptographic operations, given that such value is negligible when compared with the effort required to process security for the same 6LoWPAN packet. The labels in Figure 12 indicate the computational time required to process security for a 102-byte 6LoWPAN packet. Also note that the values illustrated in Figure 12 are total values, measured from the reception of a 6LoWPAN packet to the time when the respective cryptographic algorithm finishes processing the packet, and therefore represents the

DOI: 10.1002/dac

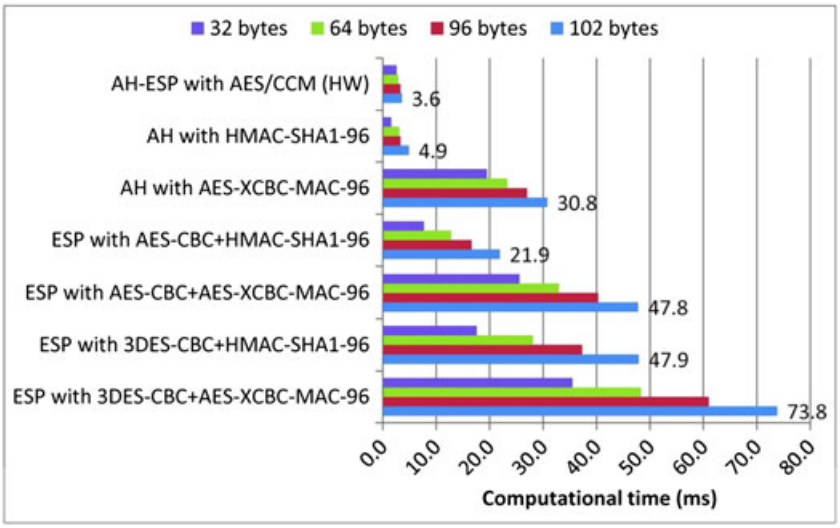

Figure 12. Computational time required by 6LoWPAN security.

total computational effort required to process security for a 6LoWPAN packet in the TelosB. The values were obtained from measurements using the 32 kHz internal oscillator of the TelosB, which is accessible to TinyOS applications via the *counter* interface.

Comparing Figures 11 and 12, as expected we are able to observe a close relationship of energy consumption and computation time. The fact that the results are not directly drawn from one another may be explained by differences in the computational efficiency inherent of each algorithm and of the software implementations employed on our experimental evaluation. From Figure 12 we observe that the most demanding cryptographic suite appropriate to ESP is 3DESCBC when used together with AES-XCBC-MAC-96, requiring in total approximately 74 ms for processing a 102-byte 6LoWPAN packet. Regarding the support of integrity and authentication, HMAC-SHA1-96 appears as the most efficient algorithm available in software. AESXCBCMAC96, although providing greater security is much more demanding, requiring approximately 31 ms to process a full-sized 6LoWPAN packet. Standalone hardware encryption appears again as the most efficient solution, and in this case the time required to process the same 6LoWPAN packet was measured as 3.6 ms. Because AES/CCM implements the CCM* combined mode, this in reality represents the time necessary to encrypt, decrypt or generate the authentication data for a 6LoWPAN packet. Regarding AES implemented in software, we observe that AES-CBC is clearly more demanding, although better than 3DES-CBC in providing confidentiality.

6. OVERALL EVALUATION OF 6LOWPAN SECURITY

Our experimental evaluation study on the resources required from constrained sensing devices to support 6LoWPAN security allows us to consider its impact in more concrete application scenarios. We therefore proceed to discuss the viability of our proposal regarding sensing applications with diverse requirements in terms of security, communication rates, and lifetime of sensing devices.

6.1. Impact of 6LoWPAN security on the communications rate of sensing devices

Because sensing applications may be very diverse in terms of the employed communications rate, we find it appropriate to evaluate if 6LoWPAN security may represent a bottleneck in this respect. This is an important evaluation aspect because, as we have seen, security introduces a non-negligible computational overhead on constrained smart objects, which are unable to process packets received or waiting transmission while the microcontroller is busy performing cryptographic operations.

DOI: 10.1002/dac

When considering communications using IEEE 802.15.4 at 250 kb/s, we realize that the impact of the computational time required for security on the maximum transmission rate is much larger than the impact on the time required for the transmission of a few more bytes required for the 6LoWPAN security headers and the MIC code. What we cannot exclude from consideration is the overhead introduced by IEEE 802.15.4 addressing on the bandwidth available for 6LoWPAN. This overhead represents 19.6% of the total bandwidth, because 25 bytes are required for link-layer information with each 127 bytes 6LoWPAN packet, as can be seen in Figure 1.

In Figure 13 we illustrate the maximum transmissions rate, which can be achieved by sensing application employing 6LoWPAN security, considering the usage of the various cryptographic suites with 6LoWPAN packets with 32, 64, 96, or 102 bytes. The values obtained and illustrated in this figure are in packets per second and are valid for AH and ESP in both tunnel and transport modes, together with the transmission of the authentication data, if required. The values illustrated in Figure 13 consider the time required for the processing of 6LoWPAN headers (including security) on the TelosB, which we have experimentally measured as 0.09 ms. We do not represent the values for the maximum transmission rate without security, but those values are fundamentally greater, in particular 252 packets per second for 102 bytes 6LoWPAN packets, 268 for 64 byte packets, 402 for 96 bytes packets and 803 for 32 bytes packets. From Figure 13 we can observe that the impact of 6LoWPAN security is particularly relevant when transmitting smaller packets. For larger packets (for example for packets measuring from 64 to 102 bytes) security still allows acceptable transmission rates, particularly using cryptographic suites based on AES/CCM and SHA1. One possible design approach for 6LoWPAN applications would therefore be to employ aggregation of sensing data whenever applicable, because this allows reducing the impact of security on the communications rate.

As illustrated in Figure 13, security configurations employing 3DES cause a greater impact on the maximum available communications rate. For applications requiring only integrity and authentication, AH using HMAC-SHA1-96 or hardware AES/CCM are good choices. HMAC-SHA1-96 appears in fact again as a superior choice in providing such security properties using a software implemented security algorithm. Again regarding authentication and integrity, HMAC-XCBCMAC-96 causes a greater impact as can be observed in Figure 13. It can be nevertheless an appropriate choice for applications requiring lower transmission rates, because it provides security superior to HMACSHA1-96. In general, we observe that acceptable transmission rates are achievable using 6LoWPAN security. Because applications are usually designed to save energy by not requiring large transmission rates, the limits identified in Figure 13 should not represent a limitative factor of the applicability of 6LoWPAN security.

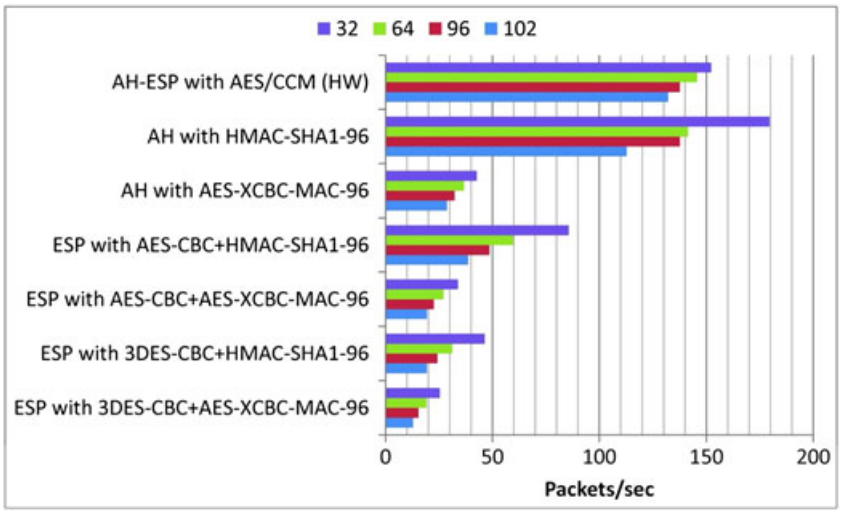

Figure 13. Maximum transmission rate with 6LoWPAN security.

DOI: 10.1002/dac

6.2. *Impact of 6LoWPAN security on the lifetime of sensing applications*

Other than the impact of 6LoWPAN security on the communication rate smart objects are able to achieve, it is also important to analyze its impact on the lifetime of such sensing devices, because it in the end may determine the lifetime of a given sensing application. The importance of this evaluation is related to the fact that most sensing applications designed for the IoT will only be viable if able to operate in unattended mode during a long period of time, because in many situations smart objects are devices for which it is difficult or impossible to replace batteries during long periods of time. As for our previous evaluation studies, our overall goal is to analyze if acceptable compromises can be achieved between the usage of resources on smart objects and security. In Figures 14 to 17 we illustrate the lifetime that a TelosB sensing device can achieve using 6LoWPAN security to process packets with different sizes and using different communication rates. In particular, we consider the usage of lower transmission rates (from 1 to 10 transmitted packets per second) and higher transmission rates (from 20 to 200 transmitted packets per second). We also consider the processing of 32 and 102 bytes 6LoWPAN packets, because this represents two complementary scenarios in terms of the size of 6LoWPAN packets processed in such communications. The achievable lifetime are represented in days for each security and usage configurations, and because of the wide range of values we use a logarithmic scale for the representation of the obtained values.

The values illustrated in Figures 14 to 17 are derived from our experimentally obtained values using a TelosB mote powered using two new AA LR6-type batteries. As for our previous evaluation, we also consider the energy required for the processing of 6LoWPAN headers in each packet (including security headers), which was experimentally measured as 0.007 nJ per 6LoWPAN processed packet with security. This value reflects the total energy required for the processing of a 6LoWPAN packet, from the invocation of the transmission of the packet using the BLIP networking stack to the time of the completion of its transmission. For comparison purposes, Figures 14 to 17 also illustrate the expected lifetime when using 6LoWPAN communications without security.

As with our study on the impact of security on the transmissions rate, we do not consider the extra energy required for the transmission of 6LoWPAN headers in tunnel mode versus transport mode, neither for the transmission of authentication data. This is due to the fact that the energetic cost of the transmission or reception per bit with the TelosB is very small, and consequently represents

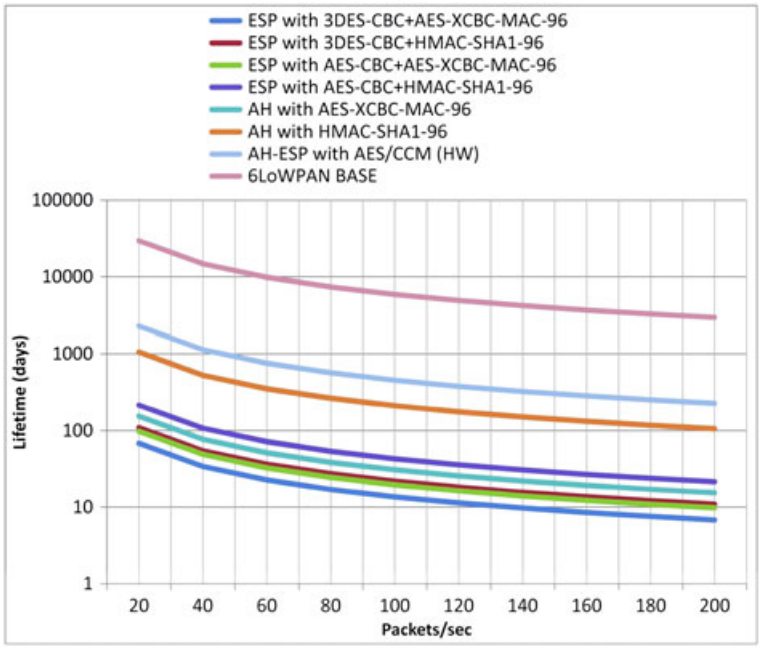

Figure 14. Lifetime of a sensing device when processing security for a 102 bytes 6LoWPAN packet, considering higher communication rates.

DOI: 10.1002/dac

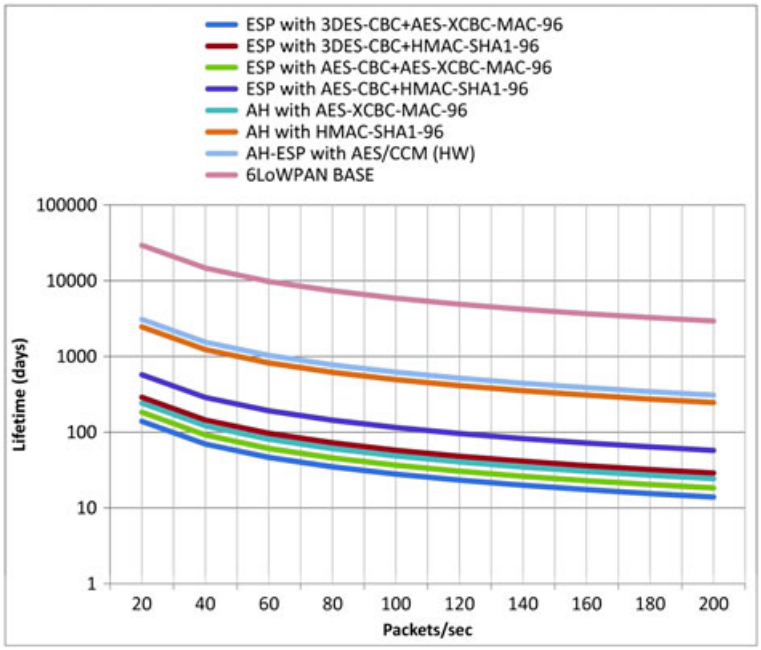

Figure 15. Lifetime of a sensing device when processing security for a 32 bytes 6LoWPAN packet, considering higher communication rates.

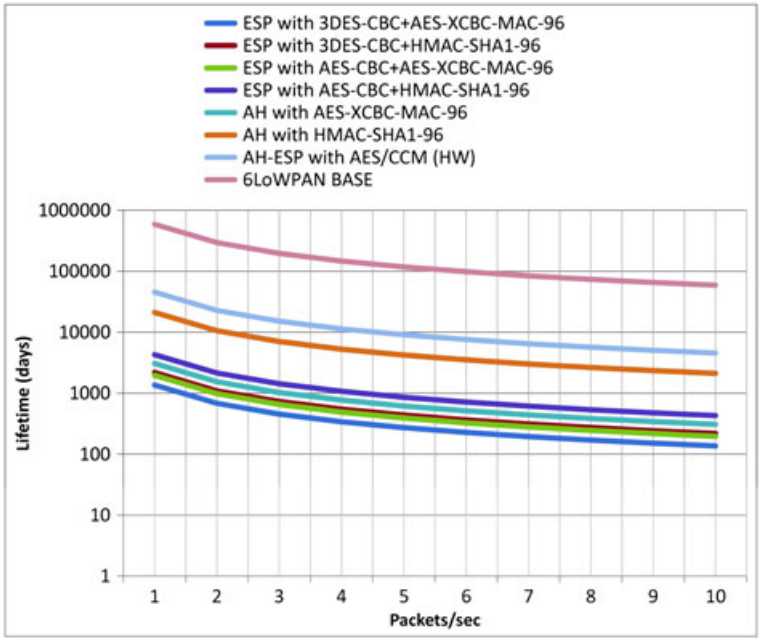

Figure 16. Lifetime of a sensing device when processing security for a 102 bytes 6LoWPAN packet, considering lower communication rates.

a negligible impact on the lifetime of the applications and does not influence our analysis and conclusions. We observe that AES-CCM and HMAC-SHA1-96 for integrity and authentication allow much higher lifetime of the sensing device, particularly for applications requiring lower transmission rates. 3DESCBC appears again as a bad choice independently of the transmission rate, while cryptographic suites employing AESCBC and XCBC-MAC-96 in software are possible

DOI: 10.1002/dac

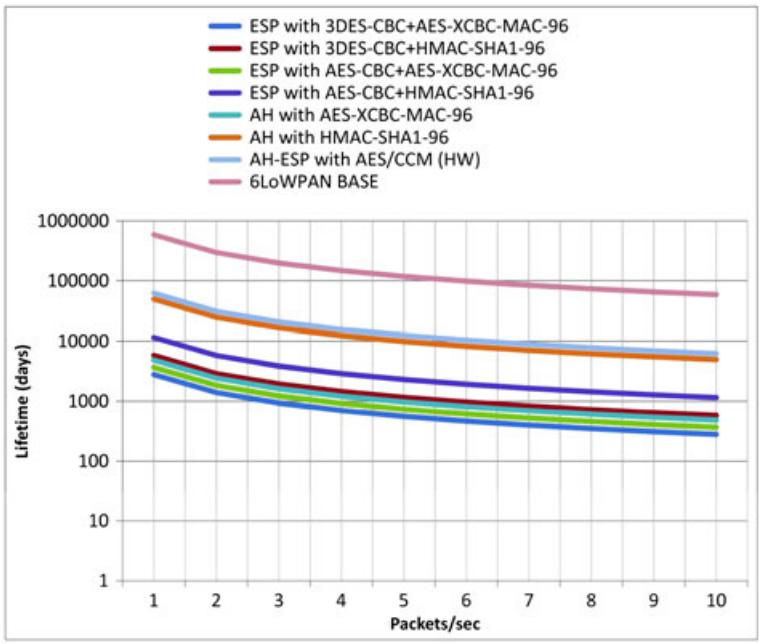

Figure 17. Lifetime of a sensing device in days when processing security for a 32 bytes 6LoWPAN packet, considering lower communication rates.

choices if applications require lower transmission rates. For the processing and transmission of smaller (32 bytes) 6LoWPAN packets, the impact of authentication and integrity using AH with HMAC-SHA1-96 is almost equal to hardware AES/CCM. HMACSHA196 can therefore be a good alternative in providing authentication and integrity for applications transmitting smaller data payloads, in particular for the usage with smart objects that do not support hardware encryption. We can observe that security introduces a non-negligible impact on the lifetime of applications if compared with the baseline usage scenario without network-layer security. Nevertheless, it can also be observed that the achievable lifetime using 6LoWPAN security is in general very good, in particular for sensing applications that require lower transmission rates. Because many applications on the IoT will probably employ lower transmission rates, we can see that in such situation the other cryptographic suites based on the usage of software AES-CBC and of AES-XCBC-MAC-96 are also viable. It is therefore perfectly possible to employ such cryptographic suites both in software and hardware (for smart objects supporting it) with 6LoWPAN while not critically impacting the lifetime of the sensing device. This factor, together with the conclusions obtained in our previous evaluation studies, allows us to enforce our conviction on the effectiveness of the usage of 6LoWPAN security in the context of an appropriate security architecture for the IoT.

7. CONCLUSIONS

The IPv6 protocol and the 6LoWPAN adaptation layer can play a major role in the evolution of the Internet as we know it today. Because the Internet embraces sensorial capabilities, new and exciting applications may come to life that will require and benefit from the availability of end-to-end network-layer communications between smart objects and other sensing devices or Internet hosts. Such communications can only be viably employed if appropriate security mechanisms are adopted. In the current paper we propose and experimentally evaluate new compressed security headers for 6LoWPAN, and such headers were designed in a way such as to ease its integration with the IP Security architecture as it evolves in the future. As we have verified, 6LoWPAN security can be viably employed in various configurations by sensing applications with

DOI: 10.1002/dac

different requirements in terms of communication rates and payload space. We show that network-layer security can be a reality for applications used in the context of the IoT. Because the proposed mechanisms allow for the usage of different security configurations, security can be adapted to the particular requisites of each application, therefore allowing the establishment of acceptable compromises between security and usage of resources on constrained sensing devices.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the EU Seventh Framework Programme (FP7/20072013) under grant agreement n° 224282, Project GINSENG. The authors also thank the Communications and Telematics Group of the CISUC-University of Coimbra.

REFERENCES

1. Atzori L, Iera A, Morabito G. The Internet of Things: A survey. *Computer Networks* 2010; **54**(15):2787–2805. DOI: 10.1016/j.comnet.2010.05.010.
2. Chen X, Makki K, Yen K, Pissinou N. Sensor network security: a survey. *IEEE Communications Surveys & Tutorials* 2009; **11**(2):52–73. DOI: 10.1109/SURV.2009.090205.
3. Roman R, Najera P, Lopez J. Securing the Internet of Things. *IEEE Computer* 2011; **44**(9):51–58. DOI: 10.1109/MC.2011.291.
4. Control W, M A. (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), IEEE std. 802.15.4, 2006.
5. Rodrigues J, Rodrigues P, Neves P. A survey on IP-based wireless sensor network solutions. *International Journal of Communication Systems* 2011; **23**(8):963–981. DOI: 10.1002/dac.1099.
6. Ko J, Terzis A, Dawson-Haggerty S, Culler D, Hui J, Levis P. Connecting low-power and lossy networks to the internet. *IEEE Communications Magazine* 2011; **49**(4):96–101. DOI: 10.1109/MCOM.2011.5741163.
7. Kushalnagar N, *et al.* IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals. *RFC 4919*, 2007.
8. Montenegro G, *et al.* Transmission of IPv6 Packets over IEEE 802.15.4 Networks. *RFC 4944*, 2007.
9. Le A, Loo J, Lasebae A, Aiash M, Luo Y. 6LoWPAN: a study on QoS security threats and countermeasures using intrusion detection system approach. *International Journal of Communication Systems* 2012; **25**:1189–1212. DOI: 10.1002/dac.2356.
10. Park S, Kim K, Haddad W, Chakrabarti S, Laganier J. IPv6 over Low Power WPAN Security Analysis.draft-daniel-6lowpan-security-analysis-05, 2011.
11. 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; 802–807, DOI: 10.1109/MAHSS.2008.4660130.
12. Granjal J, Monteiro E, Silva JS. A secure interconnection model for IPv6 enabled wireless sensor networks. *Proceedings of IFIP Wireless Days, 2010*, Venice, Italy, 2010; 1–6, DOI: 10.1109/WD.2010.5657743.
13. Granjal J, Monteiro E, Silva JS. Enabling network-layer security on IPv6 wireless sensor networks. *Proceedings of IEEE GLOBECOM 2010*, Miami, USA, 2010; 1–6, DOI: 10.1109/GLOCOM.2010.5684293.
14. 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; 1112–1117, DOI: 10.1109/WAINA.2009.47.
15. Gupta V, *et al.* Sizzle: A standards-based end-to-end security architecture for the embedded Internet. *Proceedings of the Third IEEE International Conference on Pervasive Computing for the Embedded Internet (PERCOM '05)*, Hawaii, USA, 2005; 247–256, DOI: 10.1109/PERCOM.2005.41.
16. Casado L, Tsigas P. ContikiSec: A secure network layer for wireless sensor networks under the contiki operating system. *Proceedings of the 14th Nordic Conference on Secure IT Systems: Identity and Privacy in the Internet Age (NordSec '09)*, Oslo, Norway, 2009; 133–147, DOI: 10.1007/978-3-642-04766-4_10.
17. Raza S, Duquennoy S, Chung T, Yazar D, Voigt T, Roedig U. Securing communication in 6LoWPAN with compressed IPsec. *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (DCOSS '11)*, Barcelona, Spain, 2011; 1–8, DOI: 10.1109/DCOSS.2011.5982177.
18. The Contiki OS. (Available from: http://www.contiki-org.org) [accessed July 2012].
19. Juels A. RFID security and privacy: a research survey. *IEEE Journal on Selected Areas in Communications* 2006; **24**(2):381–394. DOI: 10.1109/JSAC.2005.861395.
20. Lee C-F, Chien H-Y, Laih C-S. Server-less RFID authentication and searching protocol with enhanced security. *International Journal of Communication Systems* 2011; **25**(3):376–385. DOI: 10.1002/dac.1246.
21. Hui J, Thubert P. Compression Format for IPv6 Datagrams over IEEE 802.15.4-based Networks. *RFC 6282*, 2011.
22. Kent S, Seo K. Security Architecture for the Internet Protocol. *RFC 4301*, 2005.
23. TelosB MP. (Available from: http://www.xbow.com/pdf/Telos_PR.pdf) [accessed July 2012].
24. Kent S. IP Encapsulating Security Payload. *RFC 4303*, 2005.

25. Roman R, Alcaraz C, Lopez J, Sklavos N. Key management systems for sensor networks in the context of the Internet of Things. *Journal of Computers and Electrical Engineering* 2011; **37**(2):147–159. DOI: 10.1016/j.compeleceng.2011.01.009.
26. Kivinen T. Minimal IKEv2. draft-kivinen-ipsecme-ikev2-minimal-00.txt.
27. Housley R. Using Advanced Encryption Standard CCM Mode with IPsec Encapsulating Security Payload (ESP). *RFC 4309*, 2005.
28. Kent S. IP Authentication Header. *RFC 4302*, 2005.
29. Oliveira L, Sousa A, Rodrigues J. Routing and mobility approaches in IPv6 over LoWPAN mesh networks. *International Journal of Communication Systems* 2011; **24**(11):1445–1466. DOI: 10.1002/dac.1228.
30. The TinyOS O S. (Available from: http://www.tinyos.net) [accessed July 2012].
31. Hui JW, Culler DE. IP is dead, long live IP for wireless sensor networks. *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems (SenSys '08)*, New York, USA, 2008; 15–28, DOI: 10.1145/1460412.1460415.
32. Eastlake D. Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH). *RFC 4305*, 2005.
33. Madson C, Glenn R. The use of HMAC-SHA-1-96 within ESP and AH. *RFC 2404*, 1998.
34. Frankel S. The AES-XCBC-MAC-96 Algorithm and Its Use With IPSec. *RFC 3566*, 2003.
35. Standalone hardware AES Encryption using CC2420. (Available from: http://cis.sjtu.edu.cn/index.php/The_Standalone_AES_Encryption_of_CC2420_(TinyOS_2.10_and_MICAz)) [accessed July 2012].

AUTHORS' BIOGRAPHIES

Jorge Granjal is an Assistant Professor at the Department of Informatics Engineering of the University of Coimbra, Portugal, and a researcher of the Laboratory of Communication and Telematics of the Centre for Informatics and Systems of the University of Coimbra, Portugal. His main research interests are computer networks, network security and wireless sensor networks. He is a member of IEEE and ACM communications group. He is currently pursuing a PhD in the area of security in wireless sensor networks and the Internet of Things.

Edmundo Monteiro is a Full Professor at the University of Coimbra, Portugal, form where he obtained his PhD in Electrical Engineering, Informatics Specialty. His research interests are computer networks, wireless communications, service oriented infrastructures and security. He is author of several publications including books, patents and over 200 papers in national and international refereed books, journals and conferences. Edmundo Monteiro is the Portuguese representative in IFIP-TC6; he is member of IEEE Communications, IEEE Computer and ACM Communications groups.

Jorge Sá Silva received his PhD in Informatics Engineering in 2001 from the University of Coimbra, where is an Assistant Professor at the Department of Informatics Engineering of the University of Coimbra and a Senior Researcher of Laboratory of Communication and Telematics, Portugal. His main research interests are mobility, network protocols and wireless sensor networks. He has been serving as a reviewer and publishing in top conferences and journals in his expertise areas. His publications include two book chapters and over 70 papers in refereed national and international conferences and magazines. He is a member of IEEE and is a licensed professional engineer. His homepage is at http://www.dei.uc.pt/~sasilva.