6LoWPAN Extension for IPsec

Shahid Raza[1], Thiemo Voigt[1], Utz Roedig[2]
[1]Swedish Institute of Computer Science, Kista, Sweden
{shahid, thiemo}@sics.se
[2]Lancaster University Computing Department, Lancaster, UK
{u.roedig}@lancaster.ac.uk

***Abstract*—Real-world deployments of wireless sensor networks (WSNs) require secure communication. Recently, WSNs and traditional IP networks are more tightly integrated using IPv6 and 6LoWPAN. Available IPv6 protocol stacks can use IPsec to secure data exchange. Thus, it is desirable to extend 6LoWPAN such that IPsec communication with IPv6 nodes is possible. It is beneficial to use IPsec because the existing end-points on the Internet do not need to be modified to communicate securely with the WSN.**

We propose a 6LoWPAN extension for IPsec. Our extension supports both IPsec's Authentication Header (AH) and Encapsulation Security Payload (ESP). Thus, communication endpoints are able to authenticate, encrypt, and check the integrity of messages using standardized and established IPv6 mechanisms.

I. Introduction

Wireless Sensor Networks can be tightly integrated with existing IP based infrastructures using IPv6 over Low Power Wireless Personal Area Networks (6LoWPAN). Sensor nodes using 6LoWPAN can directly communicate with IPv6 enabled hosts. IPv6 hosts in the Internet support by default IPsec for secure communication. Therefore, if data flows between IPv6 hosts and 6LoWPAN sensor nodes it is desirable to take advantage of existing capabilities and to secure traffic using IPsec. Thus, we propose to add IPsec support to 6LoWPAN.

IPsec defines an Authentication Header (AH) [3] and an Encapsulating Security Payload (ESP) [4]. The AH can be used to provide data integrity and authentication while ESP provides data confidentiality, integrity, and authentication. Either AH, ESP or both can be used to secure IPv6 packets in transit. It is up to the application to specify which security services are required. 6LoWPAN uses header compression techniques to ensure that the large IPv6 and transport-layer headers (UDP/TCP) are reduced. By supporting IPsec's AH and ESP additional IPv6 extension headers have to be included in each packet. Thus, it is important to ensure that compression techniques are as well applied to these extension headers. The main contributions of this paper is the specification of IPsec for 6LoWPAN including definitions for AH and ESP extension headers. Prior to this work no specification for IPsec in the context of 6LoWPAN existed.

To test our IPsec extension for 6LoWPAN we extend the IP and 6LoWPAN implementation in the Contiki operating system [2]. Our results [5] clearly demonstrate that IPsec is a viable option for 6LoWPANs. On acceptance of this paper we intend to present these results in presentation.

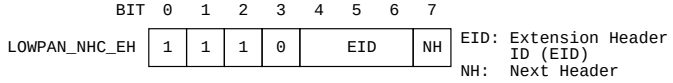

Fig. 1. LOWPAN_NHC_EH: NHC encoding for IPv6 Extension Header

II. 6LoWPAN and IPsec

6LoWPAN [1] aims at integrating existing IP based infrastructures and sensor networks by specifying how IPv6 packets are to be transmitted over an IEEE 802.15.4 network. The maximum physical-layer packet size of 802.15.4 packet is $127byte$ and the maximum frame header size is $25byte$. An IPv6 packet has therefore to fit in $102byte$. Given that packet headers of a packet would already consume $48byte$ of the available $102byte$ it is obvious that header compression mechanisms are an essential component of the 6LoWPAN standard. IPsec too requires header compression to keep packet sizes reasonable in 6LoWPAN. Unfortunately, there are no header encodings specified for AH and ESP extension headers. In this section we therefore propose these extension header encodings.

A. LOWPAN_NHC Extension Header Encoding

HC13 defines context aware header compression using IPHC for IP header compression and NHC for the next header compression. The HC13 defines the general format of NHC that starts with variable length ID bits and then contains encodings for the compressed next header. We define NHC encodings for the two IP extension header namely AH and ESP, and optionally for one newly proposed IPSEC, see Section II-D. 6LoWPAN already defines NHC encodings for IP extension headers (NHC_EH) that can be used to link AH and ESP extension headers. NHC encodings for the IPv6 Extension Headers consist of a NHC octet where three bits (bits 4,5,6) are used to encode the IPv6 Extension Header ID (EID). The NHC_EH encoding for extension headers is shown in Figure 1.

Out of eight possible values for the EID, six are specified by the HC13 draft.The remaining two slots (101 and 110) are currently reserved. We provide three proposals to use these free slots.

1) The first proposal is to use one reserve slot, say 101, to identify that the next header is IPsec header. The ID bits in the proposed NHC for AH and ESP identify that the current header is AH or ESP, see Section II-B and II-C.

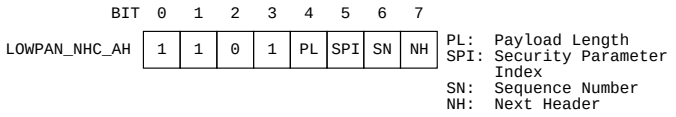

Fig. 2. NHC_AH: NHC encoding for IPv6 Authentication Header

2) In the second proposal we use one reserve slot like proposal 1. However, we define one additional NHC for IPsec, see Section II-D. This will incur an overhead of one additional byte. However, this is a flexible way that can also help to compress upper layer security protocols.
3) The third proposal is a bit cleaner but uses both reserve slots in the NHC_EH: one to encode AH and other to encode ESP. In this case we do not necessarily need ID bits in the AH or ESP because the EID field of the previous NHC_EH identifies the next header that will be either AH or ESP. However to comply with 6LoWPAN standard we set these ID bits accordingly in our proposals.

It is also necessary to set the last bit in NHC_EH to 1 to specify that the next header (AH or ESP) is encoded using NHC.

B. LOWPAN_NHC_AH Encoding

We define the NHC encoding for the AH. Our proposed NHC for AH is shown in Figure 2.

We describe the function of each header field:

- The first four bits in the NHC_AH represent the NHC ID we define for AH. These are set to 1101.
- If $PL = 0$: The payload length (length of the IPsec header) field in AH is omitted. This length can be obtained from the SPI value because the length of the authenticating data depend on the algorithm used and are fixed for any input size.
  If $PL = 1$: The payload value is carried inline after the NHC_AH header.
- If $SPI = 0$: the default SPI for the sensor network is used and the SPI field is omitted. We set the default SPI value to 1. SPI 0 is reserved to indicate that no security association exists. This does not mean that all nodes use the same security association (SA), but that every node has a single preferred SA, identified by SPI 1.
  If $SPI = 1$: All 32 bits indicating the SPI are carried inline after the NHC_AH header.
- If $SN = 0$: A 16 bit sequence number is used. The left most 16 bits are assumed to be zero.
  If $SN = 1$: All 32 bits of the sequence number are carried inline after the NHC_AH header.
- If $NH = 0$: The next header field in AH will be used to specify the next header and it is carried inline.
  If $NH = 1$: The next header field in AH is skipped. The next header will be encoded using NHC.

The minimum length of a standard AH supporting the mandatory HMAC-SHA1-96 is $24byte$. After optimal compression we obtain a header size of $16byte$. Figure 3 shows compressed IPv6/UDP packet secured with AH using HMAC-SHA1-96.

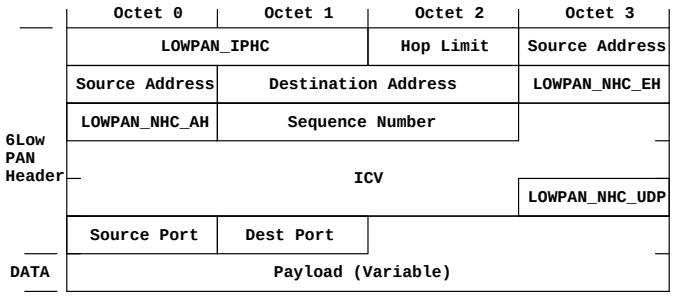

Fig. 3. Example of a compressed IPv6/UDP packet using AH

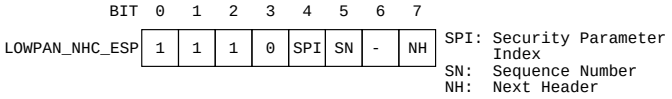

Fig. 4. LOWPAN_NHC_ESP: NHC encoding for IPv6 ESP

C. LOWPAN_NHC_ESP Encoding

Figure 4 shows the NHC encodings we propose for ESP. We describe the function of each header field:

- The first 4 bits in the NHC_ESP represent the NHC ID we define for ESP. These are set to 1110.
- If $SPI = 0$: The default SPI for the sensor network is used and the SPI field is omitted. We set the default SPI value to 0.
  If $SPI = 1$: All 32 bits indicating the SPI are carried inline after the NHC_ESP header.
- If $SN = 0$: A 16 bit sequence number is used. The left most 16 bits are assumed to be zero.
  If $SN = 1$: All 32 bits of the sequence number are carried inline after the NHC_ESP header.
- If $NH = 0$: The next header field in ESP will be used to specify the next header and it is carried inline.
  If $NH = 1$: The next header field in ESP is skipped. The next header will be encoded using NHC. This is only possible if hosts are able to execute 6LoWPAN compression/decompression and encryption/decryption jointly.

Recall that the minimum ESP overhead without authentication, AES-CBC and perfect block alignment is $18byte$. After optimal compression this header overhead is reduced to $12byte$. ESP with authentication (HMAC-SHA1-96) has an overhead of $30byte$ which is reduced to $24byte$ using the outlined ESP compression.

If ESP is used it is not possible to compress upper layer headers such as UDP. A 6LoWPAN gateway between sensor network and IP network cannot access and expand the encrypted UDP header. To enable UDP compression with ESP we need to specify a new encryption algorithm for ESP which is able to perform UDP header compression and encryption.

D. LOWPAN_NHC_IPSEC Encoding

We define the NHC encoding for the additional IPSEC. Our proposed NHC for IPSEC is shown in Figure 5.

We describe the function of each header field:

- The first four bits in the NHC_AH represent the NHC ID we define for AH. These are set to 1100.

Service	Uncompressed IPsec		Compressed IPsec		802.15.4	
	Mode	Bytes	Mode	Bytes	Mode	Bytes
AH Authentication	HMAC-SHA1-96	24	HMAC-SHA1-96	16	AES-CBC-MAC-96	12
ESP Encryption	AES-CBC	18	AES-CBC	12	AES-CTR	5
ESP Encryption and Authentication	AES-CBC and HMAC-SHA1-96	30	AES-CBC and HMAC-SHA1-96	24	AES-CCM-128	21

TABLE I
WITH COMPRESSED IPSEC, PACKET SIZES ARE SIMILAR TO 802.15.4 WHILE IPSEC PROVIDES END-TO-END SECURITY.

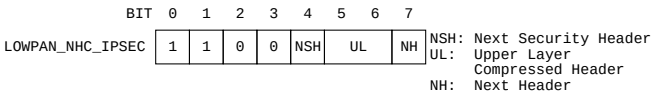

Fig. 5. NHC_IPSEC: NHC encoding for our proposed IPsec header

- If $NSH = 0$: The next header is NHC compressed AH header.
  If $PL = 1$: The next header is NHC compressed ESP header.
- UL field can take four values that can be used to define 6LoWPAN compression for UDP/TCP payload. We propose to use these bits as follows.
  - 00 : The UDP payload is NHC compressed Internet Key Exchange (IKE) header. IKE is uniquely identified by 500 or 4500.
  - 01 : The UDP or TCP payload is NHC compressed TLS or UTLS Record Protocol header, respectively. The current 6LoWPAN draft does not specify NHC for TCP.
  - Slot 10 and 11 are reserved and can be used for other upper layer compressions.
- If $NH = 0$: The next header (either AH or ESP) will be carried inline.
  If $NH = 1$: The next header will be encoded using NHC.

E. IPsec Vs. IEEE 802.15.4 Security: Packet Overhead

Currently WSN communication is secured using 802.15.4 link-layer security. This security mechanism can only provide hop-by-hop security and, in contrast to our IPsec implementation, lacks the ability to provide proper end-to-end-security. Nevertheless, we provide here a comparison of packet overheads between 802.15.4 link-layer security and IPsec security. Table I summarizes the packet overhead when using uncompressed IPsec, compressed IPsec and 802.15.4 link-layer security, with IPsec standardized algorithms.

III. CONCLUSIONS

In this paper we have given a specification of IPsec for 6LoWPAN. WSNs will be an integral part of the Internet of the future. Communication between hosts and nodes in sensor networks will be commonplace. The research community and industry agrees that IP and 6LoWPAN are the protocol standards that will be used to bring the Internet and WSNs together. IPsec is the standard method to secure IP communication and it is therefore reasonable to investigate if this mechanism can be extended to reach nodes within the WSN.

This paper shows that a compressed IPsec is a sensible and viable choice for 6LoWPANs. The key advantage of using IPsec in WSN is that we achieve *end-to-end* IP based communication between a sensor device and Internet hosts. While using IPsec, the IEEE 802.15.4 security features can be disabled as security services are provided in the IP layer.

REFERENCES

[1] G. Deloche, N. Kushalnagar, J. Hui, and D. Culler, "Transmission of IPv6 Packets over IEEE 802.15.4 Networks," RFC 4944, Internet Engineering Task Force, Sep. 2007. [Online]. Available: http://www.ietf.org/rfc/rfc4944.txt

[2] A. Dunkels, B. Grönvall, and T. Voigt, "Contiki - a lightweight and flexible operating system for tiny networked sensors," in *EMNets'04*, Tampa, USA, Nov. 2004.

[3] S. Kent, "IP Authentication Header," RFC 4302, Internet Engineering Task Force, 2005. [Online]. Available: http://tools.ietf.org/html/rfc4302

[4] ——, "IP Encapsulating Security Payload," RFC 4303, Internet Engineering Task Force, 2005. [Online]. Available: http://tools.ietf.org/html/rfc4303

[5] S. Raza, T. Chung, S. Duquennoy, D. Yazar, T. Voigt, and U. Roedig, "Securing internet of things with lightweight ipsec," SICS, Tech. Rep. T2010:08, 2010.