Lightweight IKEv2: A Key Management Solution for both the Compressed IPsec and the IEEE 802.15.4 Security

Shahid Raza, Thiemo Voigt, Vilhelm Jutvik
Swedish Institute of Computer Science, Kista, Sweden
{shahid, thiemo, ville}@sics.se

With the advent of 6LoWPAN [1], wireless sensor networks (WSN) can be connected to the traditional Internet using the well tested IP protocol. These protocols form the Internet of Things (IoT) or strictly speaking the IP-connected IoT. There is no doubt that the 6LoWPAN enabled sensors will be the underlying technology of the IoT [2]. In the IP-connected IoT, IP enabled WSN are connected to the untrusted, unreliable, and vulnerable Internet. Moreover, the wireless medium adds to untruthfulness and vulnerability. To enable secure communication with the traditional Internet one of the well tested secure Internet protocols such as IPsec should be extended to WSN.

Network or higher layer security and link layer security are not interchangeable. Upper layers are used to provide end-to-end security whereas link layer security controls access to the wireless medium. It is important to detect data modification attacks as early as possible in the wireless medium. With upper layer security protocols such as IPsec the authenticity of the data is verified at the end nodes as IPsec is end-to-end. For this reason link-layer security with authentication services is important even though we already use IPsec at the network layer.

On of the hardest problems in IoT security is the key management. IPsec uses the Internet Key Exchange (IKE) protocol [3] for manual or automatic key exchange. Manual key exchange is based on symmetric pre-shared key for the initial authentication of two communicative parties. Automatic key exchange, on the other hand, uses asymmetric cryptography for the initial authentication. Both manual and automatic key exchange protocol are mandatory in IKE and hence full-fledged Internet nodes are capable of handling both mechanisms. Manual key exchange methods are lightweight but not scalable. Moreover, they are less secure whereas automatic methods are a bit heavier but more flexible, scalable. They do not require that a machine on the traditional Internet has pre-installed symmetric keys of the device with whom it wants to communicate securely.

1	1	0	1	SPI	ET	ID	NH

Figure 1. NHC encodings for the IKE Header

Security at both the network and the link layer requires two key management protocols. IKE is a heavyweight protocol and full fledged IKE is not suitable for resource constrained nodes in the IP-connected Internet of Things (IoT). We propose a lightweight 6LoWPAN compressed IKEv2 that can be used with our previously implemented and evaluated 6LoWPAN compressed IPsec [4]. Further, we propose a solution that enables the use of IKEv2 for key management for IEEE 802.15.4 link layer security.

1 Proposed 6LoWPAN Compression for IKEv2

To cope with the size limitations of the IEEE 802.15.4 link layer frames we compress the IKE header at the 6LoWPAN layer. The IKE header is part of the UDP payload. The header that preceding UDP should specify that the next header is UDP or IKE_UDP. The 6LoWPAN compression defines NHC encodings for UDP. Unlike other specified NHCs, NHC for UDP has no provision to specify that the next header is compressed. In other words, the *NH* field is missing in the UDP encodings. This does not allow us to link the compressed IKE header with the previous compressed headers (in this case the compressed UDP header). To overcome this limitation we propose that 6LoWPAN enabled devices should be able to recognize UDP-IKE (UDPIKE) as a compressed version of UDP that contains IKE header as UDP payload. In this case, IKEv2 is always in compressed form. The compressed IKEv2 header is recognized by the unique ID bits 1101 (see Figure 1).

Figure 1 shows an NHC encodings for the IKE header where SPI, exchange types, Message ID, and Next Header fields are compressed.

- The first 4 bits in the NHC_IKE represent the NHC ID we define for the UDPIKE header. These are set to 1101.
- 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 a default

IETF/IAB workshop on Smart Object Security, March 23, 2012, Paris, France
Copyright 2012 - Shahid Raza

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 64 bits indicating the SPI are carried inline after the NHC_IKE header.

- If $ET = 0$: A 2 bit exchange type is used that can specify the four standardized exchange types.

  If $ET = 1$: All 8 bits of the exchange are carried inline after the NHC_IKE header.

- If $MessageID = 0$: A 16 bit sequence number is used. The left most 16 bits are assumed to be zero.

  If $MessageID = 1$: All 32 bits of the sequence number are carried inline after the NHC_IKE header.

- If $NH = 0$: The next header field in IKE header will be used to specify the next header and it is carried inline.

  If $NH = 1$: The next header field in IKE is skipped. The next header will be encoded using NHC. This enables us to provide 6LoWPAN NHC encodings for the IKEv2 *payloads*.

The Length field in the IKE header is emitted as it can be inferred from the lower layer. The other field are carried inline as they contain values that MUST be included in all messages. The same 6LoWPAN NHC mechanisms can be used to compress other IKEv2 *payloads*.

Recently, a generic compression scheme was proposed for the protocol header or header-like structure [5]. We also plan to make use of this scheme for the IKEv2 payloads compression, as an alternative approach.

2 IKEv2 for IEEE 802.15.4 Security

IKEv2 [3] uses the security association payload to negotiate attributes of a Security Association (SA). Each SA payload has one or more proposals for the particular SA. With other fields each proposal has a protocol ID field that indicates the protocol identifier for the current negotiation. The protocol ID is used to specify that this SA is being established for a particular security protocol. The protocol ID is 8 bits long where the three values IKE (1), AH (2), ESP (3) are already defined. We propose to use, say 4, as protocol ID for the IEEE 802.15.4 link layer security. This will allow us to establish an SA for the link layer security between two neighboring sensor nodes that are one hop apart. A sensor node in the wireless network can access the security association database (SAD) to retrieve the key and other algorithms' details to provide security at the link layer. We will provide implementation details in the upcoming release of our source code and in a subsequent publication.

3 ECC for IKEv2

IPsec uses the Diffie Hellman protocol for key exchange. The current IKE implementations for the traditional Internet mostly use RSA for automatic key exchange. However, RSA is very heavyweight for the resource constraint WSN. To overcome this we plan to use Elliptic Curve Cryptography (ECC) as an asymmetric cryptographic system in Diffie Hellman. Our implementation is the first ECC implementation for the well matured uIP stack and Contiki OS [6]. We use standardized ECC algorithms [7] and NIST recommended elliptic curve and prime numbers in our ECC implementation.

4 References

[1] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282 (Proposed Standard), September 2011.

[2] J. Vasseur and A. Dunkels. *Interconnecting Smart Objects with IP - The Next Internet*. Morgan Kaufmann, 2010.

[3] C. Kaufman, P. Hoffman, Y. Nir, and P. Eronen. Internet key exchange protocol version 2 (ikev2), sep 2010.

[4] S. Raza, S. Duquennoy, T. Chung, D. Yazar, T. Voigt, and U. Roedig. Securing Communication in 6LoWPAN with Compressed IPsec. In *Proceedings of the 7th IEEE International Conference on Distributed Computing in Sensor Systems (IEEE DCOSS 2011)*, Barcelona, Spain, June 2011.

[5] C. Borman. 6LoWPAN Generic Compression of Headers and Header-like Payloads. draft-bormann-6lowpan-ghc-03, October 2011.

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

[7] D. Fu and J. Solinas. IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA). RFC 4754 (Proposed Standard), January 2007.