An FPGA Based Reconfigurable IPSec ESP Core suitable for IoT applications

Muzaffar Rao, Joseph Coleman and Thomas Newe

Optical Fibre Sensors Research Centre,
Mobile & Marine Robotics Research Centre,
Department of Electronic and Computer Engineering
University of Limerick
Limerick, Ireland

Abstract— **This work implements an FPGA (Field Programmable Gate Array) based reconfigurable IPSec ESP core. The IPSec protocol, developed by the IETF (Internet Engineering Task Force) in 1998, is a popular solution to facilitate protection of the data being transferred at the IP layer. IPSec ESP is one of the two main IPSec protocols (AH: Authentication Header and ESP: Encapsulation Security Payload). IPSec ESP is used to provide data confidentiality security services with Authenticity (optional). Implementation of the IPSec is a computing intensive work, that's why hardware implementation of IPSec is a best solution. Here, to design IPSec ESP core an encryption algorithm AES is used. Proposed design also supports ESP-tunnel and ESP-transport mode of operation. This core is tested by applying default length of 576 bytes for an IPv4 datagram and results are reported on Virtex-5 and Virtex-6 FPGAs. The proposed IPSec ESP core can be used to provide data confidentiality security to IoT applications.**

Keywords- FPGA, AES, IPSec, ESP;

I. Introduction

One of the challenges identified by the International Telecommunication Union (ITU) in their report on "The Internet of Things" [1] is that of privacy and security. When we start being surrounded by these smart objects that are moving around, gathering information concerning our lives, behavior or habits, there will be great concern regarding the security of that information. The IPSec reconfigurable core presented here can be used to satisfy demand of security requirement of IoT devices.

In a public Internet Protocol (IP) [2][3][4] network data flow between IoT applications can be visible to any number of nodes on the network. Although data can be secured using encryption and data alteration on the network, whether the data is encrypted or not, can be detected using integrity checking. it is often the case that the facility/ability to offer these services is not available on the IoT device itself. This lack of available security makes IoT applications susceptible to various security attacks [5]. In order to address this security issue in networked communication systems the IPSec protocol [6] was developed.

IPSec is not a single protocol, but a set of protocols and services that provide a framework to implement a complete security solution for an IP network. IPSec framework provides protection for any higher-layer TCP/IP application or protocol without the need for additional security methods. The security protection is provided via adding authentication and encryption functionality into the IP packet. The contents of the authentication and encryption functionality are determined by the use of cryptographic algorithms.

The purpose of IPSec is to provide various security services to traffic travelling between a source and destination, the destination/source may be a router, or a host. The services may be applied to all packets, or only to specific types of traffic. Figure. 1 below show conceptually the protection provided by IPSec between two hosts. The line in red shows IPSec implemented on the path between Router 1 and Host B. IPSec may only operate on certain types of data while other data is transmitted on an unprotected path as shown by the black links. There may be separate IPSec protected links between the two routers and between Host B and Router 1.

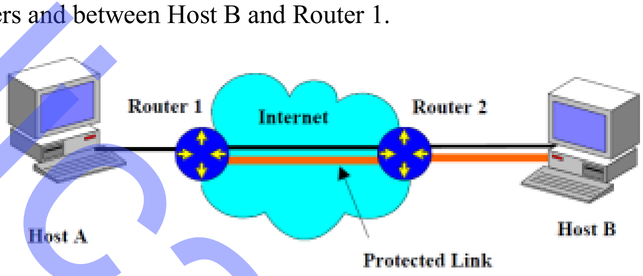


Fig. 1. Protection provided by IPSec between two hosts

In IPSec, there are two protocols used to provide security services; a) Authentication header (AH) [7], and b) Encapsulating Security Payload (ESP) [8]. The AH protocol provides support for data authentication and data integrity verification. The ESP protocol defines mechanisms for both data confidentiality and authentication (optional). Both IPSec protocols (AH & ESP) support two modes of operation, (a) Transport mode and (b) Tunnel mode. In transport mode, only the upper-layer protocol data segment of the IP datagram is authenticated and it is typically used for end-to-end protection of IP datagram packets between two hosts. In tunnel mode, the entire original IP datagram is authenticated within a new outer IP header. The Tunnel mode can be used between security gateways to create a VPN (virtual private network).

The IPSec protocol is almost always embedded into the TCP/IP protocol stack via software in the OS (operating system), such as in Linux and NetBSD. However, IPSec has proven to be computationally intensive [9], which greatly affect the performance of the network it is implemented on. Data throughput in core routers has already achieved up to terabits per second, and line card interface speeds exceed 10Gbps, yet high performance internet security device speeds are far behind these data throughputs. The main reason for this reduction in speed is that data processing requirements for security protocols is often complex and time consuming, so it is difficult for security devices to achieve equal performance when compared to internet devices. Given that software solutions to complex problems like IPSec generally suffer from low performance issues, when compared to hardware, it is necessary for high data throughput speeds that hardware implementations of IPSec are utilized.

The flexible architecture and high performance features [10] of FPGA make it suitable to use particularly for applications involving complex cryptographic algorithms. It can be said that FPGAs combine the best parts of ASICs and processor-based systems but are in fact parallel in nature. The advantage of using a software programmed processor is that software is very flexible to change while a disadvantage is that performance can suffer if the clock is not fast. The advantage of an ASIC is that it can provide very high performance because of its dedicated type of operation and its disadvantages are: 1) high cost to volume ratio; 2) extended delay between design to end product; 3) incapability to include new changes after the system is fabricated and 4) difficulties in debugging errors. FPGAs fill the gap between hardware and software and offer numerous advantages [11]. Given this the authors consider that an FPGA is the best reconfigurable hardware platform for the implementation of cryptographic algorithms

A possible implementation of an FPGA based IPSec core is suggested in Fig. 2. This is a BITW (Bump in the Wire) architecture for IPSec. In Fig. 2 two networks that previously communicated using an insecure IP link with each other can now communicate securely by layering IPSec underneath regular IP using an FPGA based BITW IPSec hardware solution. This technique allows legacy IPv4 hardware to implement IPSec without having to replace expensive networking devices.

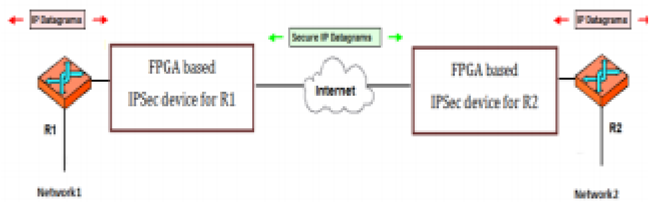


Fig. 2. FPGA based IPSec core implementation

In this work we have presented implementation of ESP protocol of IPSec that supports tunnel and transport mode of operations. Manual configuration option is used for selection of operational modes and configuration of Key. To provide encryption, AES [12] is implemented as this is the most popular encryption algorithm.

The remainder of this article is organized as follows. A brief overview of the IPSec ESP protocol is given in section 2. Section 3 discusses briefly the AES, while section 4 provides the implementation details of IPSec ESP and section 5 concludes give performance results. A section 6 concludes and references are given in section 7.

II. IPSEC ESP OVERVIEW

One of the weaknesses of the original Internet Protocol is that it lacks any sort of general purpose mechanism for ensuring the authenticity and privacy of data as it is passed over the internetwork. As IP datagrams usually be routed between two devices over unknown networks, any information in datagrams is subject to being intercepted and even possibly changed. With the increased use of the Internet for critical applications, security enhancements were needed for IP. To this end, a set of protocols called IP Security or IPSec was developed.

The IPSec technology is the one that brings secure communication to the internet protocol. The solution was designed to be usable for both IPv4 and IPv6. The IPSec is not a single protocol, but rather a set of services and protocols that provide a complete security solution for an IP network. The IPSec provides various types of services like

- Network-level peer authentication
- data origin authentication
- data integrity
- data confidentiality (encryption)
- replay protection

As mentioned in Section 1, IPSec uses two protocols to provide security. The 'AH' protocol Provides support for (1) Data integrity of an IP datagram: Because of data integrity check, modification to an IP datagram in transit is not possible (2) Authentication of an IP datagram: Because of this feature, end system can verify the sender and prevents address spoofing attacks (3) Replay protection: Guards against replay attacks. The ESP protocol provides both encryption and authentication (optional). These two protocols support following two modes of operation:

- Transport mode
- Tunnel mode

Transport mode is used for an end to end communication between two hosts, while tunnel mode provides virtual tunnel between two gateways. The IPv4 header before applying 'ESP' is given in Fig. 3.

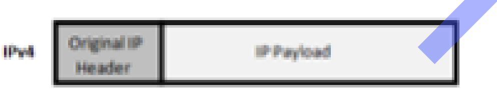
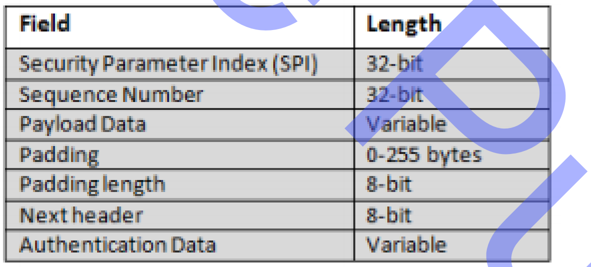


Fig. 3. IPv4 header (before applying 'ESP')

The 'ESP' protocol consists of 03 fields (1) ESP header (2) ESP trailer, and (3) ESP authentication data. The ESP header contains two fields namely; SPI and sequence number. The 'SPI' (Security Parameter Index) field identifies a security association (SA) that specifies shared security attributes between entities. According to [8], SPI value of zero is reserved for local, implementation-specific (in absence of security association). The 'Sequence number' field represents a monotonically increasing counter value that is used to provide protection again replay attack. These two fields come before the encrypted data and its placement depends on whether ESP is used in tunnel mode or transport mode. The ESP trailer is placed after the encrypted data and it contains padding (used to align the data), payload length and next header field. The authentication data field (optional) contains the integrity check value. In ESP, encryption covers payload data, padding, padding length and next header fields. The ESP structure is shown in Table 1.

Table 1. Encapsulation Security Payload (ESP)



Field	Length
Security Parameter Index (SPI)	32-bit
Sequence Number	32-bit
Payload Data	Variable
Padding	0-255 bytes
Padding length	8-bit
Next header	8-bit
Authentication Data	Variable

In transport mode encryption is provided directly between two hosts, encryption protection covers the payload of an IP datagram, while header of the IP datagram is not encrypted. In IPv4 ESP header is inserted after the IP header and ESP trailer is placed at the end of the IP datagram. The ESP insertion scheme in transport mode is shown in Fig. 4.

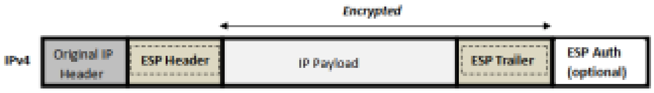


Fig. 4: ESP insertion scheme transport mode

In tunnel mode a new IP header is attached and entire datagram is treated as a payload of a new IP datagram. The new IP packet is tunneled from one gateway to another. The ESP header is inserted after the new IP header and ESP trailer is placed at the end of the packet. The ESP insertion scheme in tunnel mode is shown in Fig. 5.

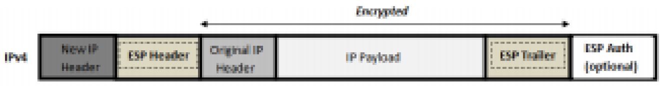


Fig. 5. ESP insertion scheme for tunnel mode

III. Advanced Encryption Standard (AES)

AES is a symmetric block cipher algorithm that processes fixed data of 128-bit blocks. Block cipher means that the number of bytes that it encrypts is fixed i.e 16 bytes. It supports key sizes of 128, 192 and 256 bits with iterative rounds of 10, 12 and 14 respectively. The number of these rounds is chosen depending on the key size. A separate Key expansion unit is used to generate keys for each round of AES algorithm. The bit series related to the input, the output and the cipher key are processed as arrays of bytes; called State. The State array consists of four rows of bytes, and every row consists of 4 bytes. In each round of AES, a 128-bit data block is transformed by a sequence of operations as given in [12]. Our earlier presented AES implementation technique [13], which used BRAM and LUTs resources of FPGA, is used in AES Counter mode here to provide required encryption for IPSec ESP core.

IV. Proposed Implementation of IPSec ESP

The presented IPSec ESP implementation involves transport and tunnel mode of operations as shown in Fig. 6. This reconfigurable core is capable to secure IPv4 datagram.

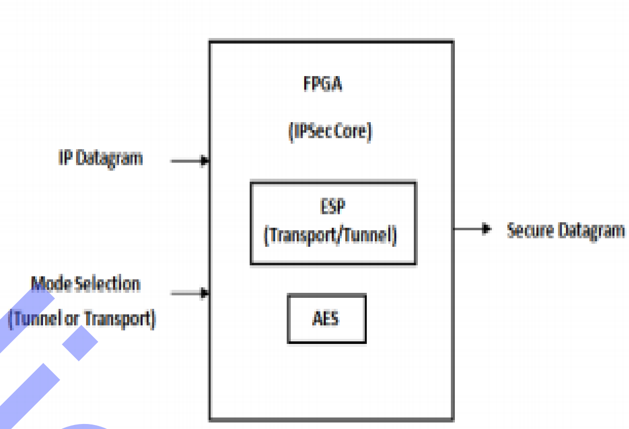


Fig. 6. Proposed Reconfigurable IPSec ESP Core

This reconfigurable IPSec ESP core is manually configurable to operate in desired mode of operation. The implemented scheme steps are presented in Fig. 7. Initial 04 bits of datagram packet are used to check the IP version. This check is necessary to have the idea about the structure of datagram. So, that different datagram's fields can be accessed and updated during ESP processing. The IP header bits are extracted from the IP datagram. Once the IP version is verified, the datagram packet is passed through the packet filter, that is used to decide either ESP processing is required or not. This packet filtering worked as security policy for IPSec. Using this packet filtering, datagram packet is dropped, forwarded without applying IPSec or forwarded with IPSec processing. In this implementation we checked 'protocol' field of IP datagram, if this field is equal to '50' (protocol number of ESP protocol), it means received datagram is already secure. In this case datagram is forwarded without applying ESP protocol. Similarly, by applying check on IP address of a particular network, IP datagram can be dropped and forwarded with and without applying ESP protocol. Next block of Fig. 7 involves selection of mode of operation that is configured through input. The last step is of main ESP protocol processing

The ESP protocol covers both, encryption and authentication (optional). In this work we have implemented encryption functionality of ESP. The ESP processing is given in Fig. 8 and it is straight forward as compared to AH processing [14] as it doest not involve pre and post protocol processing.

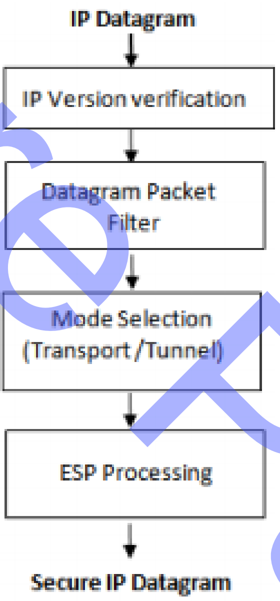


Fig. 7. Implementation steps of IPSec ESP

A. *Extract IP Payload bits from datagram*

IP Payload bits are extracted from datagram, because these bits are used in ESP header and also need to be encrypted. In IPv4, length of these extracted fields is equal to ‘ Total length – Header length’.

B. *Formation of ESP header / Trailer*

The ESP protocol fields are detailed in Table 1. The ‘SPI’ field is set to zero, which indicates that no security association exists. The ‘sequence number’ field is generated by using 32-bit unsigned counter, which increases by one whenever ESP header is generated. The sequence number of first secure datagram packet is 1. The ‘Padding’ field of ESP trailer is used here to make sure the length of plain text required by AES core for encryption. The ‘Padding length’ field of ESP trailer is set to the number of bytes inserted in padding field. In transport mode the ‘Next header’ field of ESP header is set to ‘Protocol’ field of IPv4 header. In tunnel mode the ‘Next header’ field of ESP represents the encapsulated IP datagram, that’s why it is set to ‘4’ where 4 is the protocol number of IPv4.The extracted IP payload is inserted between ESP header and ESP trailer in transport mode, while in tunnel mode complete IP datagram is added between ESP header and trailer.

C. *Perform Encryption*

The proposed implementation of AES referred in section 3 is used to apply encryption service. The encryption coverage depends on selected mode of operation. For transport mode IP payload and ESP trailer is encrypted and for tunnel mode complete IP datagram and ESP trailer is encrypted. The AES core is used in counter mode, so that a number of 128-bit blocks encryption can be done in paralle.

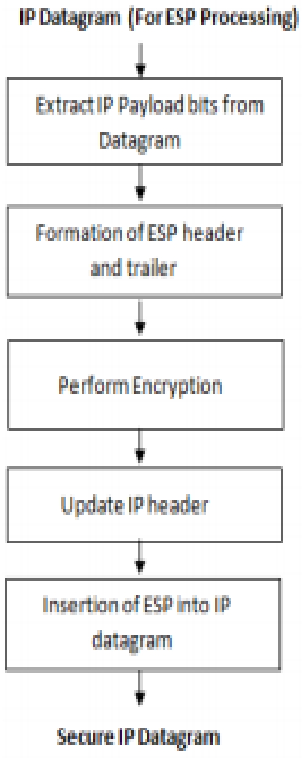

IP Datagram (For ESP Processing)
Extract IP Payload bits from Datagram
Formation of ESP header and trailer
Perform Encryption
Update IP header
Insertion of ESP into IP datagram
Secure IP Datagram

Fig. 8. ESP Processing steps

D. *Update IP header*

In IPv4 two fields of header are updated (1) Protocol (2) Header checksum. The ‘Protocol’ field is set to the ‘Protocol’ number of ESP i.e 50 for both transport and tunnel mode. Because of this change a new header checksum is calculated and is placed to the ‘Header checksum’ field of IPv4 header.

E. *Insertion of ESP into IP datagram*

The last step in ESP processing is insertion of ESP into IP datagram, and it depends on selected mode of operation and IP version. This scheme is implemented as mentioned in Fig. 4 and Fig. 5. Now, the secure IP datagram is ready to forward.

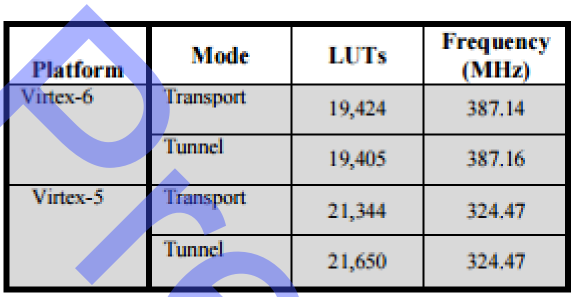
V. PERFORMANCE RESULTS

In this work a Virtex-5 and a Virtex-6 FPGA is used for the implementation of IPSec ESP protocol. Selection of these Xilinx FPGAs was made because of their high performance feature. The design was implemented and synthesized using ISE Xilinx 14.2 tool and the HDL language Verilog is used.

The target device xc5vtx240t-2ff1759 was used for Virtex-5 and xc6vcx75t-2ff484 for Virtex-6. The ESP core is designed to support a default length of 576 bytes for an IPv4 datagram.

Table 2 provide results for ESP-transport and ESP-tunnel mode of operation. These results shows that our proposed scheme for IPSec ESP can be used for high data throughput applications.

Table 2. IPSec ESP Processing results for IPv4 datagram



Platform	Mode	LUTs	Frequency (MHz)
Virtex-6	Transport	19,424	387.14
	Tunnel	19,405	387.16
Virtex-5	Transport	21,344	324.47
	Tunnel	21,650	324.47

VI. Conclusion

In this work an FPGA based reconfigurable IPSec ESP core implementation is presented that can be used to provide security services to IoT applications using BITW IPsec solution. This BITW implementation will help to introduce IPSec security features in existing IPv4 network. The proposed implementation takes IPv4 datagram and an additional header (ESP header) is inserted in datagram. This implementation support both, transport and tunnel mode of operation and these modes can be configure manually for the selection of specific mode of operation. The cryptographic function, AES is implemented to provide encryption services for ESP protocol. The ESP protocol implementation mainly involves verification of version, packet filter, extraction of IP payload bits from datagram, formation of ESP header/Trailer, Encryption of selected IP datagram fields, formation of updated IP header and insertion of ESP header into IP packet.

Acknowledgment

The authors would like to thank the Erasmus Mundus STRoNGTiES (Strengthening Training and Research through Networking and Globalization of Teaching in Engineering Studies) program for providing funding that has facilitated the completion of this work. In addition this work was supported in part by SFI-Science Foundation Ireland under Grant No. SFI/12/RC/2302.

References

[1] ‘The internet of Things’. Technical report, International Telecommunication Union, 2005. ISBN: 92-61-11291-9.

[2] “RFC-791”, http: // www . ietf . org / rfc / rfc791 . txt.

[3] “RFC-1349”, http : // www . ietf . org / rfc / rfc1349 . txt.

[4] “RFC-2474”, http : // www . ietf . org / rfc / rfc2474 . txt.

[5] M. Healy, T. Newe and E. Lewis. “Security for Wireless Sensor Networks: A Review”. IEEE Sensors Applications Symposium (IEEE SAS 2009) February 17th–19th, 2009. New Orleans, LA, USA. pp 80-85. ISBN: 978-1-4244-2787-1.

[6] S. Kent, and R. Atkinson,“Security architecture for the internet protocol”, IETF network working group, RFC2401, 1998.

[7] “RFC-4302”, https : // www . ietf . org / rfc / rfc4302 . txt.

[8] “RFC-4303”, https: // www. ietf.org / rfc / rfc4303. txt. Platform Mode LUTs Frequency (MHz) Virtex-6 Transport 19,424 387.14 Tunnel 19,405 387.16 Virtex-5 Transport 21,344 324.47 Tunnel 21,650 324.47

[9] A. Ferrante,V.Piuri, and J. Owen, “IPSec Hardware Resource Requirements Evaluation”, Next Generation Internet Networks (NGI 2005), April 2005. pp.240-246, “DOI:10.1109/NGI.2005.1431672”.

[10] W. Vander, K. Benkrid, “High-Performance Computing Using FPGAs”, Springer book, ISBN: 978-1-4614-1790-3.

[11] National Instruments. Introduction to FPGA Technology: Top Five Benefits. http : // zone. ni. Com / devzone / cda / tut / p / id/6984, December 2010.

[12] NIST, “Advanced encryption standard (aes), fips 197,” November 2001. [Online]. Available: http : // csrc .nist. gov / publications / fips / fips197 / fips-197 . pdf.

[13] Rao M., Newe T. and Grout I., “AES implementation on Xilinx FPGAs suitable for FPGA based WBSNs”. 9th International Conference on Sensing Technology (ICST 2015), 08 Dec - 10 Dec 2015, Auckland, New Zealand.

[14] Rao M., Newe T. and Grout I., Lewis E., Mathur A. “FPGA Based Reconfigurable IPSec AH Core Suitable for IoT Applications ”. 13th IEEE International Conference on Pervasive Intelligence and computing (PICom - 2015), Liverpool, United Kingdom.