FPGA implementation of IPsec protocol suite for multigigabit networks

Mateusz Korona[1], Krzysztof Skowron[1], Mateusz Trzepiński[1], Mariusz Rawski[1]
[1] Instytut Telekomunikacji, Wydział Elektroniki i Technik Informacyjnych, Politechnika Warszawska, Nowowiejska 15/19, 00-665, Warszawa, Poland
rawski@tele.pw.edu.pl

***Abstract* - IPsec is a suite of protocols that adds security to communications at the IP level. However, the high computing power required by the IPsec algorithms limits network connection performance. The paper presents the hardware implementation of IPsec gateway in FPGA. Efficiency of the proposed solution allows to use it in networks with data rates of several Gbit/s**.

***Keywords* - IPsec, FPGA, hardware implementation**

I. INTRODUCTION

Along dynamic development of the Internet, it turned out that global web is vulnerable to many vicious attacks. Data are transmitted through multiple routers and, as a result, are easy to intercept (*packet sniffing*). Active attack methods, consisting of packet forging, are also used (*packet spoofing*, *man in the middle*). Economic globalization raised a demand for safe communication mechanisms for international companies and protection of profits generated by e-business, which prompted creation of proper safety mechanisms.

Despite the emersion of solutions applied in higher layers of OSI reference model (like SSL/TLS or SSH), the work on IPsec, the protocol suite providing security within third layer of OSI RM, was started. Its main advantage is a "transparence" for the higher layers. It works with different network protocols and enables easy switching between cryptographic algorithms (in an event of safety breach). With IPsec, it's possible to create VPNs (*Virtual Private Networks*) suited for aforementioned e-business needs.

There exists a variety of available IPsec solutions. *Open source* software implementations are suitable only for private use (e.g. Linux OpenWRT distribution [1], designed for home routers, which offers throughput of the order of several dozen Mbit/s). More efficient solutions require significant hardware resources (over a dozen processor cores) to achieve maximal throughput of 1 Gbit/s (strongSwan project [2][3]). Commercial implementations often use hardware support (e.g. Intel IPsec library designed for Intel Architecture processors with extended instruction set [4]). These solutions offer higher throughputs (of the order of several Gbit/s, [5][6]), but are expensive. There are commercial implementations of IPsec using FPGAs [7] or GPUs [8], but they don't provide all IPsec features (e.g. IP traffic header processing or any IPsec database functions).

This paper describes the idea and implementation of IPsec gateway in FPGA, which offers all features of pure IPsec protocol suite and is efficient enough to be used in networks with data rates of several Gbit/s. Design does not support NAT-T (Traversal) feature and if it is located behind NAT, separate UDP encapsulation for ESP protocol must be performed.

Section II. presents fundamentals of IPsec protocol suite. In section III. prototype of IPsec gateway is described and possible areas of its improvement are mentioned. Section IV discusses the architecture of completely functional IPsec gateway and its performance is analyzed in section V. Section VI concludes the paper.

II. IPSEC STRUCTURE

IPsec and especially its subprotocol ESP (*Encapsulating Security* Payload) provides confidentiality, integrity and authenticity of transmitted data. Tunnel operation mode allows for creation of VPN networks.

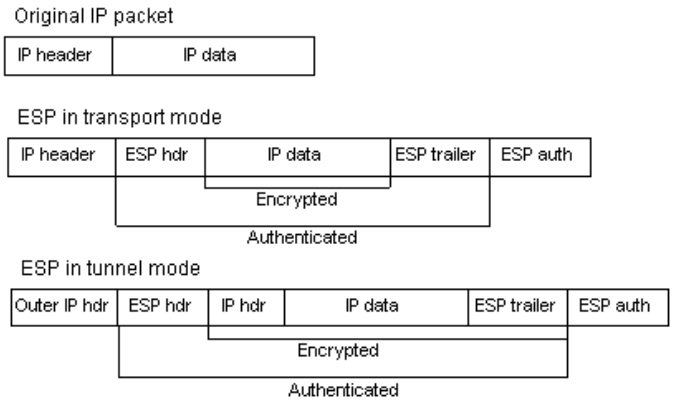

Figure 1. IP Packet secured with ESP protocol (transport and tunnel mode of operation). Source: *amaranten.com*

In tunnel mode (Fig. 1) original packet is padded, so the block cipher may be applied (length of the message must be multiple of cipher's block length). Padding contains also information arising from the structure of the ESP header (*ESP trailer*). After the padding operation, the packet is encrypted. Then ESP header (*ESP hdr*) is attached. It contains Security Parameter Index – a value that identifies safe channel SA (*Security Association*) through which the packet is transmitted.

978-1-5090-6344-4/17/$31.00 ©2017 IEEE

Thanks to SPI, destination gateway is able to identify proper cryptographic algorithm and keys to decrypt the packet. Finally, ICV (*Integrity Check Value*) is appended (*ESP auth*, e.g. *keyed-hash Message Authentication Code* – HMAC hash). Recalculating this value allows to verify integrity and authenticity of the packet. Finally, outer IP header is attached.

Fig. 2 presents the architecture of IPsec system. *Security Policy Database* is the implementation of corporate security policy. SPD stores selectors (e.g. IP addresses, port numbers, etc.), which are the basis for deciding, if and how each particular packet should be protected. *Security Association Database* stores information necessary to secure the packet (e.g. the encryption key). *Internet Key Exchange* module is able to automatically set up new safe channels (*Security Association*) and module marked as IPsec is responsible for all cryptographic operations and packet encapsulation.

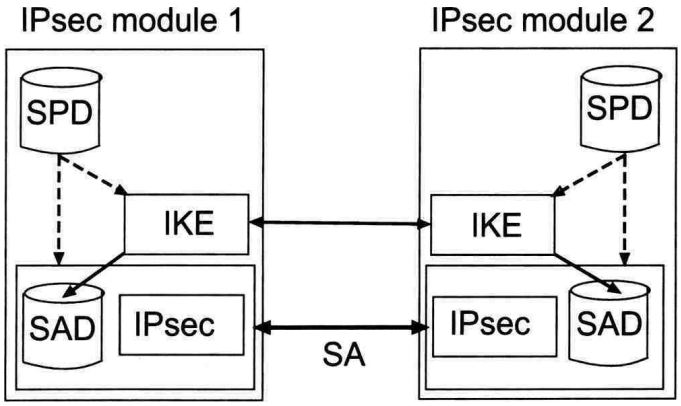

Figure 2. Architecture of IPsec system. Source: *codeidol.com*

III. PROTOTYPE DESIGN

A prototype of the IPsec gateway was designed [9], that maximizes benefits of hardware packet processing, by implementing all stages in FPGA, in contrast to other solutions that use hardware only as accelerator for critical functions. Design was partitioned into data plane and software control plane (including IKE module).

A. Prototype implementation

Fig. 3 presents the scheme of packet-securing part of the design (AST is abbreviation of AXI4-Stream interface). After the reception, IP packets are moved to simple security policy controller. Appropriate records are found in SPD database, basing on Pearson's hash calculated from IP header selectors. This module determines whether data need to be secured or not (*bypass* path). The securing data process is carried out by *IPsec_outbound* module. Subsequent processing steps are: SAD database lookup, ESP encapsulation and DES encryption. The last step is HMAC-SHA1 hash calculation. Packet is stored in FIFO, until the last calculation is finished.

B. Prototype performance analysis

Prototype of IPsec gateway was tested with Terasic DE2-70 developer board equipped with Altera's Cyclone II FPGA. Maximum clock frequency of the design was 86,72 MHz with 20% FPGA logic utilization. Based on formula

$$R_b\left[\frac{Mbit}{s}\right] = \frac{8*N\left[\frac{byte}{packet}\right]}{M\left[\frac{cycles}{packet}\right]} * F\ [MHz] \quad (1)$$

where N – size of the packet, M – number of cycles necessary for packet processing, F – maximum clock frequency of the design, throughput of the IPsec gateway for several packet lengths was computed (Tab. 1).

TABLE 1. IPSEC GATEWAY PROTOTYPE PERFORMANCE

Packet size [byte]	Cycles/packet	R_b [Mbit/s]
64	520	85.39
240	794	209.70
1500	2682	437.44

The correctness of prototype operation was verified with functional simulation and tests including packet generator and sniffer (Ostinato and Wireshark respectively). Prototype performance analysis allowed to identify three key elements critical for throughput increase.

The main "bottleneck" of the prototype is the module responsible for HMAC hash calculation. HMAC algorithm requires that SHA-1 digest is computed twice – for 240-byte IP packet it takes 86% time of its processing (685 cycles of total 794).

Another problem is the design of security processor – IPsec functions are divided between security policy controller and complicated module performing all other operations. During HMAC hash calculation stage, which is the longest one in the whole process, resources responsible for IP and ESP encapsulation or data encryption remain unused.

On the other hand, low FPGA logic utilization (20%) of small by today's standards Cyclone II FPGA allows for

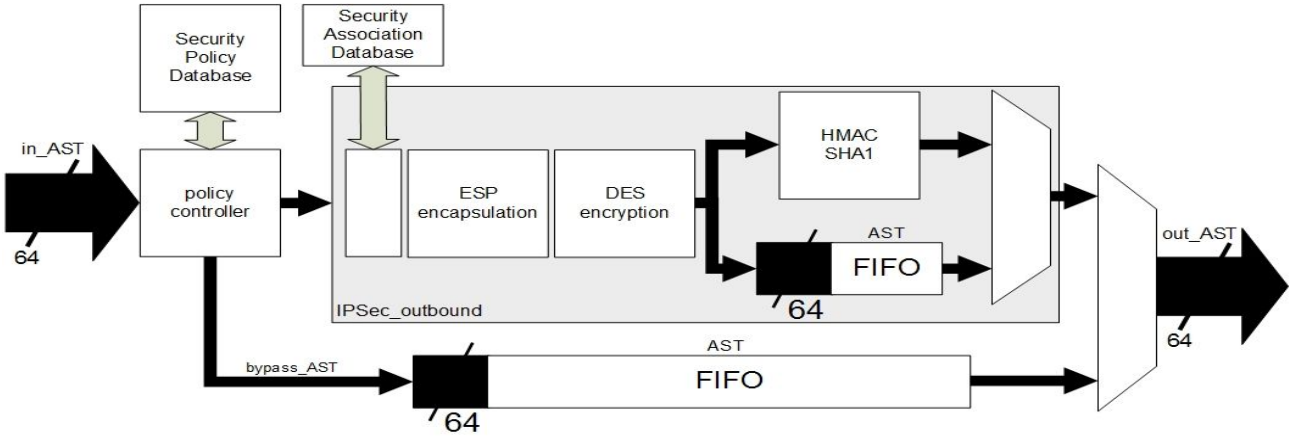

Figure 3. Block diagram of the prototype

application of hardware optimization methods, such as pipelining and parallelization of the processing.

IV. IPSEC GATEWAY

The prototype analysis became the basis for development of new hardware IPsec gateway architecture. Advanced hardware optimization methods were applied for its implementation, leading to increase in processing speed and better logic utilization (Fig 4).

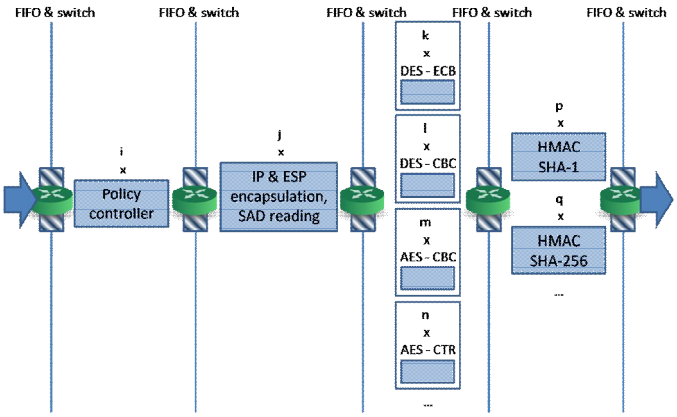

Figure 4. IPsec gateway architecture

The IPsec functions are divided into several sections of the design. Every section can contain many modules implementing a specific stage of processing, working in parallel. Sections are separated with FIFO queues and *round robin* switches, which distribute packets between modules in particular section. As a result, multiple packets can be processed at the same time. The utilization of switches enables controlling the load of processing elements in given section. Increasing the number of modules calculating the most time-consuming HMAC resulted in throughput increase. The advantage of presented architecture is also the simplified application of many different cryptographic algorithms (e.g. several ciphers in various modes in ciphering section).

A. HMAC-SHA1 module optimization

In order to improve performance of IPsec gateway, optimizations of individual modules were also performed. Special attention was devoted to HMAC-SHA1 module, because its efficiency was the crucial processing speed curtailment in the prototype. Fig. 5 presents HMAC algorithm [10].

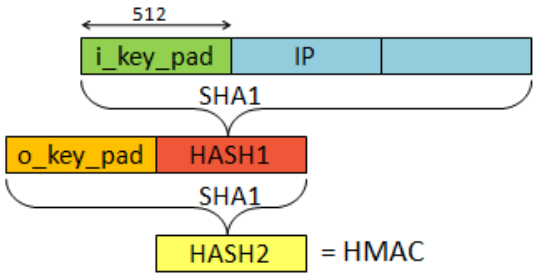

Figure 5. HMAC-SHA1 algorithm scheme

SHA-1 hash is calculated from IP packet with prepended *i_key_pad* value. String of bits is split into 512-bit blocks, which are processed individually. In the second stage, *o_key_pad* value is prepended to previously calculated hash and the whole procedure is repeated. Finally, 160-bit HMAC hash is obtained, ready to be appended to the secured packet. *I_key_pad* and *o_key_pad* blocks are the result of XOR operation performed on the key and two constants defined in RFC 2104 [10].

The same key is used for all IP packets being secured by one SA channel. Therefore, SHA-1 hashes of blocks *i_key_pad* and *o_key_pad* can be calculated in advance and SHA-1 module only has to be initialized properly. This optimization shortens the HMAC calculation by 160 clock cycles.

Another important task is reduction of time required for 512-bit block processing in SHA-1 algorithm [11]. Unfortunately, hardware algorithm's loop unrolling (e.g. calculation of 2 of 80 rounds in one clock cycle) results in the extension of the design's critical path and in the decrease of its maximum frequency.

The solution of this problem was proposed in paper [12]. SHA-1 algorithm was modified by adding new variables. Implementation of the improved algorithm decreased the time of the 512-bit block processing to 40 clock cycles (from the initial 80) without causing a drop in the design's maximum frequency.

B. Architecture optimizations

The width of the internal bus was increased from 64 to 128 bits (size of the AES block) in order to improve the performance. Two parallel DES pipelines (64-bit width) were placed in DES-ECB ciphering modules.

The use of many parallel processing cores created a problem in providing access to *Security Association Database*. To solve this, appropriate arbiter system was designed. Moreover, the accesses to the database were moved to a separate clock domain, because the DDR memory can usually be clocked with much higher frequency than other design parts.

V. PERFORMANCE ANALYSIS

The design was tested with new Terasic DE5-Net developer board equipped with Altera's Stratix V FPGA. Synthesis of the IPsec gateway in various configurations (with varying number of particular modules in the gateway section) was conducted. Tab. 2 presents synthesis results for DES configurations. In the *Configuration* column, numbers of security policy controllers, SAD controllers, ciphering modules and HMAC modules in every section were listed respectively.

978-1-5090-6344-4/17/$31.00 ©2017 IEEE

TABLE 2. DESIGN IMPLEMENTATION RESULTS FOR DES CONFIGURATIONS

Configuration	F_{max} [MHz]	Logic utilization
4/4/4/4	102.79	16%
4/4/4/8	104.04	20%
8/8/8/16	100.65	38%

The same throughput estimation method as in the prototype was used to compute the throughput of every configuration. It must be noted that benefits of the new architecture are visible when packet *stream* is being processed. In tables 3, 4 and 5, throughput (Mbit/s) for miscellaneous packet streams is presented.

TABLE 3. THROUGHPUT FOR 64-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/4/4	1137.91	1253.06	1319.84	1355.97
4/4/4/8	1691.06	2029.28	2254.75	2387.38
8/8/8/16	1940.06	2780.86	3550.16	4120.05

TABLE 4. THROUGHPUT FOR 240-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/4/4	2363.55	2526.17	2616.16	2663.61
4/4/4/8	3699.20	4342.54	4756.11	4979.33
8/8/8/16	4218.24	5784.79	7103.89	**8018.07**

TABLE 5. THROUGHPUT FOR 1500-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/4/4	3239.63	3403.22	3478.30	3526.33
4/4/4/8	6139.47	7168.50	7732.62	8081.76
8/8/8/16	6185.03	7975.65	9291.33	**10227.44**

Similar tests were conducted for IPsec gateway using AES block cipher in CBC mode, which is more advanced than DES. Because the most time-consuming part in this configuration is the ciphering process, the number of ciphering modules in section was increased. Tab. 6 presents synthesis results for AES configurations, while design throughput for miscellaneous packet streams is shown in tables 7, 8 and 9.

TABLE 6. DESIGN IMPLEMENTATION RESULTS FOR AES CONFIGURATIONS

Configuration	F_{max} [MHz]	Logic utilization
4/4/6/4	94.07	27%
8/8/12/8	97.98	34%

TABLE 7. THROUGHPUT FOR 64-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/6/4	830.411	973.01	1064.39	1116.84
8/8/12/8	1254.14	1658.37	1976.98	2187.06

TABLE 8. THROUGHPUT FOR 240-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/6/4	1711.98	1990.24	2166.29	2266.53
8/8/12/8	2537.90	3355.57	4056.53	4424.76

TABLE 9. THROUGHPUT FOR 1500-BYTE PACKETS

Configuration	Count of packets in stream			
	16	32	64	128
4/4/6/4	2598.65	2755.95	3032.10	3076.89
8/8/12/8	3362.10	4580.40	4961.40	**5518.80**

VI. CONCLUSION

This paper presents hardware implementation of IPsec gateway in FPGA. Thanks to the use of advanced designing methods and the optimization typical for digital circuits implementation, high processing speed was achieved. This allows to apply the proposed solution in networks with data rates of several Gbit/s. Presented results prove that proposed implementation can compete with commercial solutions, especially these implemented in FPGA [7]. It is important to emphasize, that current logic utilization level of Stratix V FPGA allows to expand existing IPsec gateway configurations and improve the throughput.

In view of the fact that Google along with CWI Amsterdam has recently announced breaking information about generating SHA-1 hash collisions, IPsec gateway team is currently working on hardware implementation of safer hash algorithms such as SHA-256 to prove, how easily one cryptographic algorithm may be replaced with another using described gateway architecture.

REFERENCES

[1] *IPsec performance*, OpenWRT project webpage (accessed 17.05.16), https://wiki.openwrt.org/doc/howto/vpn.IPsec.performance

[2] strongSwan project webpage, (accessed 17.05.16), https://www.strongswan.org/

[3] Klassert Steffen, 2010, "Parallelizing IPsec", https://www.strongswan.org/docs/Steffen_Klassert_Parallelizing_IPsec.pdf

[4] Intel Corporation, 2012, "Fast Multi-buffer IPsec Implementations on Intel Architecture Processors"

[5] Cisco Systems, Inc., 2008, "Cisco IPsec and SSL VPN Solutions Portfolio"

[6] Juniper Networks, 2015, "Security Products Comparison Chart"

[7] Helion Technology Limited, "IPsec ESP IP Core for FPGA - Product Brief", (accessed 19.05.16), http://www.heliontech.com/IPsec.htm

978-1-5090-6344-4/17/$31.00 ©2017 IEEE

[8] Sangjin Han, Keon Jang, Kyoung Soo Park, Sue Moon, 2010, "PacketShader: a GPU-accelerated Software Router", http://shader.kaist.edu/packetshader/

[9] Mateusz Korona, 2015, "Implementation of IPsec protocol suite using field-programmable devices", bachelor thesis

[10] Krawczyk H., Bellare M., Canetti R., RFC 2104, 1997, "HMAC: Keyed-Hashing for Message Authentication"

[11] Eastlake D. 3rd, Jones P., RFC 3174, 2001, "US Secure Hash Algorithm 1 (SHA1)"

[12] Eun-Hee Lee, Seok-Man Kim, Chungbuk National University, "Design of High Speed SHA-1 Architecture Using Unfolded Pipeline for Biomedical Applications", (accessed 26.05.15), http://www.iiis.org/CDs2009/CD2009SCI/SCI2009/PapersPdf/S231IM.pdf

978-1-5090-6344-4/17/$31.00 ©2017 IEEE