HARDWARE IMPLEMENTATION OF THE IMPROVED WEP AND RC4 ENCRYPTION ALGORITHMS FOR WIRELESS TERMINALS

Panu Hämäläinen, Marko Hännikäinen, Timo Hämäläinen, and Jukka Saarinen
Digital and Computer Systems Laboratory, Tampere University of Technology
Hermiankatu 3 A, FIN-33720 Tampere, FINLAND
Tel: +358 3 365 2111, Fax: +358 3 365 4575
e-mail: panu.hamalainen@tut.fi, marko.hannikainen@tut.fi, timo.d.hamalainen@tut.fi, jukka.saarinen@tut.fi

ABSTRACT

This paper presents hardware implementations for Improved Wired Equivalent Privacy (IWEP) and RC4 ("Ron's Cipher #4") encryption algorithms. IWEP is a block algorithm providing light-strength encryption. The algorithm has been designed for a new Wireless Local Area Network (WLAN), called TUTWLAN (Tampere University of Technology Wireless Local Area Network). On the contrary RC4, developed by RSA Data Security, Inc., is a powerful stream algorithm used in many commercial products. It is also utilized in the Wired Equivalent Privacy (WEP) standard algorithm for WLANs. The objective of this work has been to study the suitability of hardware implementation for these previously software-implemented ciphers. Hardware is needed to replace software especially in wireless multimedia terminals, in which real-time data processing and limited on-chip memory sizes are key elements. The implementations are made in Very high-speed integrated circuit Hardware Description Language (VHDL) on Xilinx Field Programmable Gate Array (FPGA) chips.

1 INTRODUCTION

Wireless local area network (WLAN) technology is seen very promising for various types of wireless indoor and limited outdoor communications, such as ad hoc networking and for continuous access to company network servers. WLANs provide network solutions when wired networks become impossible or inconvenient. Typical examples are networks that are set up on temporary basis and fixed networks in buildings in which cabling is not reasonable.

Even though wireless connections make networks very flexible and more convenient, they also bring an important issue into closer consideration: data security. The reason to this is that WLANs are far easier to eavesdrop compared to the traditional cable networks. At its simplest level, data is available to anyone within the range of a transmitting device. Therefore, these networks are very sensitive to security violations and need powerful encryption. On the other hand, because of the real-time requirements and limited processing capabilities of small terminals (e.g. on-chip memory), encryption methods used should also be efficient and simple. To avoid exceeding processing limits without losing the reliability of encryption, it would be profitable to implement the algorithms in low-cost hardware, and thereby more capacity would be left to user applications.

In this paper the Xilinx FPGA (Field Programmable Gate Array) implementations of two encryption algorithms, Improved Wired Equivalent Privacy (IWEP) and RC4 ("Ron's Cipher #4") are presented. The first algorithm (i.e. cipher) was developed at Tampere University of Technology (TUT) [1]. It was designed considering hardware implementations for the light encryption of time-critical data in TUTWLAN system [2]. Therefore, the results shown here are very promising. On the contrary, RC4's software-oriented design leads to results nearly equal to its software implementations. This cipher was developed by RSA Data Security, Inc. and is used in several commercial products.

The paper is organised as follows. At first, the IWEP algorithm and implementation results are introduced, followed by RC4 implementation. Discussion of the results is aroused in the concluding section.

2 IMPROVED WIRED EQUIVALENT PRIVACY (IWEP) IMPLEMENTATION

High level illustration of IWEP is presented in Figure 1. Despite its name the cipher is considerably different from Wired Equivalent Privacy (WEP) algorithm included into IEEE 802.11 standard for WLANs [3]. The WEP standard utilizes RC4 stream cipher. However, IWEP is a block cipher.

IWEP encrypts data in 64-bit blocks and uses a 64-bit key. The data ciphering is firmly based on permutation and exclusive-or (XOR) operation. The encryption consists of three iteration rounds, every round including a XORing and a permutation round. In the encryption and decryption operations, both the data block and the key are divided into eight 8-bit subblocks used in the iteration rounds. More information about the algorithm can be found in [1].

The IWEP implementation consists of three parts: single XOR item, XORing round (includes four XOR items), and

IWEP itself (includes three XORing rounds and data permutations). Encryption and decryption blocks are almost similar: the differences are inside the single XOR item and in addition the data permutations are done backwards. As both the encryption and decryption result into almost the same amount of gates, only encryption is discussed here.

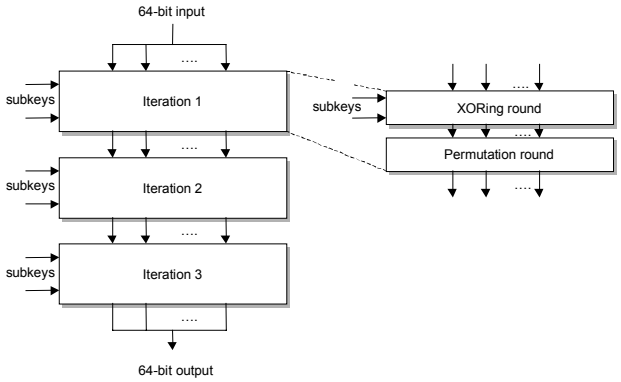

Figure 1. High level structure of Improved WEP.

The implementation was made in Very high-speed integrated circuit Hardware Description Language (VHDL) on Xilinx XC4000E [4] series chips. The used software was a PC version of Xilinx Foundation F1.5 [5]. Parts that fit to the XC4000E-4010EPQ160-2 were tested on the chip while others were only simulated. The on-chip testing was made on TUTWLAN demonstrator platform [6] using HP 16702A pattern generator/logic analyzer [7, 8].

1.1 Single XOR item

The single XOR item was implemented as shown in Figure 2. Each input and output is an 8-bit subblock of the actual data. The output is achieved one clock cycle after setting up the inputs.

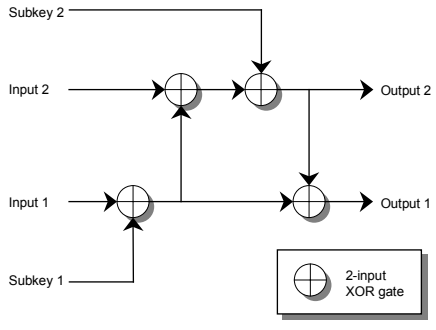

Figure 2. A Single XOR item.

This entity is very fast. It was estimated to work properly with up to 93-MHz clock on the used FPGA chip. More detailed results given by Xilinx Foundation are presented in Table 1.

Table 1. The implementation results of a single XOR item on Xilinx XC4000E-4013EPQ208-2 chip.

Quantity	Value
External I/O Buffers	49/160 (30%)
Configurable Logic Blocks (CLBs)	8/576 (1%)
CLB Flops	0/1,152 (0%)
4-input Look Up Tables	16/1,152 (1%)
Gate Count	192
Maximum Frequency	93.629 MHz
Data Throughput with Max. Freq.	187 MB/s
Data Throughput with 40-MHz clock of the TUTWLAN platform	80.0 MB/s

1.2 XORing round

The XORing round includes four parallel XOR items described above. Each item is fed with two 8-bit parts of the 64-bit data and similarly with two 8-bit parts of the 64-bit encryption key (see Figure 3). Since in one XORing round there are only four XOR items, the output here is also achieved after single clock cycle.

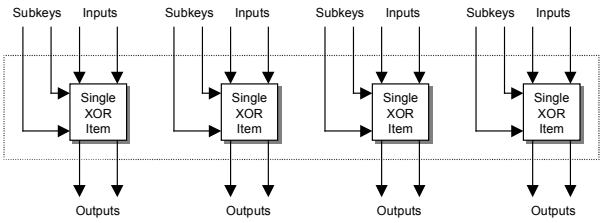

Figure 3. The implementation of one XORing round.

The single XORing round block was estimated to be able to work properly with 92 MHz. The slight decrease in efficiency compared to a single XOR item is due to the fact that four times larger logic was produced (32 CLBs). In both cases the critical path is the same. In this case the key input was excluded because of the limited amount of the input pins on the FPGA.

1.3 IWEP

The highest layer of the implementation is the IWEP module itself. In this module there are three layers of XORing rounds as defined earlier. The signals between the layers are connected according to the desired permutation.

Since it takes one clock cycle for each XORing round to produce its output, the whole encryption for the *first* message block takes three clock cycles. However, because there are registers after each XORing round it is possible to pipeline the process. Applying this means that a new data block can be fed to the cipher on every clock tick and (after the first three clock cycles) a new ciphertext block is received on every clock tick. Therefore, it can be estimated that encryption/decryption with IWEP takes only one clock cycle per block. The pipeline is depicted in Figure 4.

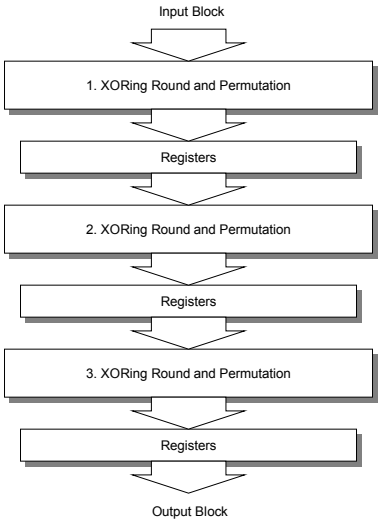

Figure 4. IWEP pipeline.

Table 2 presents the IWEP implementation results. As shown in the table, it was estimated that encryption will work properly with up to 86-MHz clock. Also here the key was held constant.

Table 2. The implementation results of IWEP on Xilinx XC4000E-4013EPQ208-2 chip.

Quantity	Value
External I/O Buffers	129/160 (80%)
Configurable Logic Blocks (CLBs)	64/576 (11%)
CLB Flops	128/1,152 (11%)
4-input Look Up Tables	96/1,152 (8%)
Gate Count	1,728
Maximum Frequency	86.081 MHz
Data Throughput with Max. Freq.	690 MB/s
Data Throughput with 40-MHz clock of the platform	320 MB/s

Because each XORing round in IWEP module is exactly similar, it was possible to implement the 64-bit cipher with only one XORing layer by adding a state machine that feeds the layer with correct inputs (implements the permutations). This way the module was expected to get smaller and more space on the chip was assumed to be saved for other applications. However, this result was not reached. Instead the implementation took more space (2166 gates), and the maximum clock frequency was markedly decreased down to 36 MHz.

Since the maximum clock frequency of the pipeline implementation is so high, one more improvement could possibly be achieved by removing the registers between the iteration rounds. This way the whole encryption/decryption could be executed during one clock cycle. This would increase also the throughput of short messages to the level of larger messages (i.e. only one clock cycle per one 64-bit block). The clock frequency should still be close to the target of 40 MHz.

3 RC4 IMPLEMENTATION

RC4 is a stream cipher, which means that it operates plaintext a single byte at a time. The high level structure of RC4 is shown in Figure 5. The algorithm consists of two main elements. In the figure, *F* denotes a key dependent function that initializes and mixes up the 256-byte memory array called substitution box (S-box). *Index* refers to the memory address and *data* to the stored or read byte.

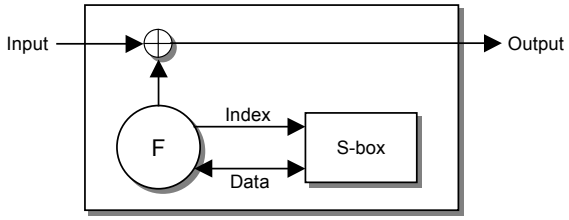

Figure 5. RC4 high level illustration.

The main idea of the algorithm is to produce an 8-bit pseudo random number series initialized by the given key. An encrypted output is then achieved by XORing the random bytes with the input data bytes. In this cipher the encryption and decryption methods are exactly similar and therefore the presented implementation can be used in either way. More detailed information on the algorithm can be found in [9].

RC4's FPGA implementation consists of three modules: a 256-byte memory array and two state machines. The memory array (S-box) is used for producing a pseudo random number stream. The state machines are needed for performing the initialization and mixing up of the array. The implementations were made using the same chips and tools as in IWEP implementation.

1.4 Substitution Box

A Xilinx design tool LogiBLOX [5] was used for implementing the S-box. The result was a 256-byte RAM module with asynchronous read operation and synchronous write operation. By using the tool it was possible to take advantage of the FPGA chip's internal RAM blocks. To test the functionality of the memory module, it was implemented and simulated without any other logic. The area and timing results are presented in Table 3.

Table 3. The implementation results of the substitution box on Xilinx XC4000E-4013PQ208-2.

Quantity	Value
External I/O Buffers	25/160 (15%)
Configurable Logic Blocks (CLBs)	88/576 (15%)
CLB Flops	0/1,152 (0%)
4-input Look Up Tables	176/1,152 (15%)
3-input Look Up Tables	72/576 (12%)
32X1 RAMs	64
Gate Count	8,516
Maximum Frequency	48.955 MHz

1.5 RC4 Cipher

The encryption function of the algorithm was implemented with two separate state machines: one for initializing the S-box and the other for the encryption itself. On the top level these modules were attached to the S-box to generate the cipher. The outcome of the resulting logic for the complete cipher is shown in Table 4. The presented throughput with 40 MHz is only theoretical because the attained maximum frequency is much lower.

Table 4. The implementation results of RC4 cipher on Xilinx XC4000E-4013EPQ208-2 chip.

Quantity	Value
External I/O Buffers	148/160 (92%)
Configurable Logic Blocks (CLBs)	255/576 (44%)
CLB Flops	142/1,152 (12%)
4-input Look Up Tables	444/1,152 (38%)
3-input Look Up Tables	103/576 (17%)
32X1 RAMs	64
Gate Count	11,372
Maximum Frequency	17.744 MHz
Data Throughput with Max. Freq. (after setup)	2.22 MB/s
Data Throughput with 40-MHz clock (after setup, theoretical)	5.00 MB/s

To find out if the results could be improved, the state machines were combined into a single unit. In fact, improvement was achieved in the number of reserved CLBs. However, this approach did not lead to any faster implementation because the maximum clock frequency was slightly decreased. Table 5 presents the results.

Table 5. The implementation results of RC4 with one state machine on Xilinx XC4000E-4013EPQ208-2 chip.

Quantity	Value
External I/O Buffers	148/160 (92%)
Configurable Logic Blocks (CLBs)	224/576 (38%)
32X1 RAMs	64
Gate Count	10,653
Maximum Frequency	16.940 MHz
Data Throughput with Max. Freq. (after setup)	2.12 MB/s
Data Throughput with 40-MHz clock (after setup, theoretical)	5.00 MB/s

4 DISCUSSION

As stated before, IWEP's hardware-oriented design made the implementation simpler and more efficient than that of RC4. Therefore, IWEP was considered very suitable for hardware implementations and especially for WLAN applications with limited processing capacity (only one clock cycle per 64 bits of data). However, IWEP is not very reliable against brute-force attack, and therefore it can only be used to encrypt time-critical data, i.e. data that is useful only for a short period of time. On the other hand, the design of IWEP makes the cipher very flexible for reconfiguration. Modifications, like changing of the permutations, altering the subkey and subdata schedule, and adding iteration rounds, can be easily done. With these frequent changes the unpredictability and consequently the security level of the cipher is increased.

Unlike IWEP, RC4 is commonly regarded as a very powerful cipher. Therefore, it is a more suitable alternative for long-term privacy. However, the cost of this improved security is the longer encryption time. Especially the initialization of the cipher is time-consuming. One solution is to implement the RC4's S-box without using a separate memory block with address and data buses, or to use RAM with also asynchronous write operation. This way the several read and write operations of the algorithm do not have such a dominating role.

REFERENCES

[1] K. Salli, T. Hämäläinen, J. Knuutila, J. Saarinen, "Security Design for A New Wireless Local Area Network TUTWLAN", The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'98), September 8-11, 1998, Boston, USA, pp. 1540-1544.

[2] M. Hännikäinen, J. Knuutila, A. Letonsaari, T. Hämäläinen, J. Jokela, J. Ala-Laurila and J. Saarinen, "TUTMAC: A Medium Access Control Protocol for A New Multimedia Wireless Local Area Network", The Ninth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC'98), September 8-11, 1998, Boston, USA, pp. 592-596.

[3] IEEE P802.11 D5.0, "IEEE Standard for local and metropolitan area networks: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications", IEEE Standards Department, 19.7.1996.

[4] "XC4000E and XC4000X Series Field Programmable Gate Arrays", Product Specification, Xilinx, November 1997, USA.

[5] "Foundation Series 1.5i Install and Release Document", Xilinx, December 1998, USA.

[6] M. Kari, M. Hännikäinen, T. Hämäläinen, J. Knuutila and J. Saarinen, "Configurable Platform for a Wireless Multimedia Local Area Network", The 5th International Workshop on Mobile Multimedia Communications (MoMuC'98), October 12-14, 1998, Berlin, Germany, pp. 301-306.

[7] "HP 16522A 200-M Vectors/s Pattern Generator User's Guide", Publication 16522-97005, Hewlett-Packard Company, August 1997.

[8] "HP 16557D 135-MHz State/500-MHz Timing Logic Analyzer User's Reference", Publication 16557-97000, 1st edition, Hewlett-Packard Company, March 1998.

[9] B. Schneier, "Applied Cryptography: Protocols, Algorithms and Source Code in C", 2nd edition, John Wiley & Sons, Inc., 1996, USA, 758 pages.