Issues in Wireless Security Based on AES Hardware Implementation

Slobodan Bojanic, Carlos Carreras, Juan M. Díez, and Octavio Nieto-Taladriz,

Abstract— While short-term IEEE 802.11 securitv solution (TKIP) wireless accommodates existing hardware, the bngterm security solution called CCMP is targeted at new hardware designs. This paper related to FPGA/ASIC hardware is architectural options in implementation of required AES crypto algorithm. We are addressing different design criteria like highspeed, low-cost, or modes of operation like CCM. These issues are also of interest for wireless sensor networks whose secure lowpower operations are referred by another IEEE standard (802.15.4).

Index Terms—AES, FPGA, security, wireless

1. INTRODUCTION

EEE standard 802.11-1999 has the ability to handle wireless traffic quite easily, but its WEP (wireless equivalent protection) security scheme has been flawed. Consequently IEEE 802.11 Task Group i (TGi) offered the security solution in two phases: short-term security solution called Temporal Key Integrity Protocol (TKIP) and long-term security solution called Counter Mode with CBC-MAC Protocol (CCMP). TKIP accommodates existing hardware and fixes all known WEP vulnerabilities, but provides only minimal security on deployed equipment and degrades performance [1].

CCM is a generic authenticate-andencrypt block cipher mode. It provides authenticated encryption combining Counter (CTR) mode encryption and CBC-MAC authentication using a single key. It assumes 128 bit block cipher - IEEE 802.11i uses AES. It is intended for packet environments with no attempt to accommodate streams. This method proves to be viable for several reasons: there is no known patent encumbrances, the modes have been used and studied for a long time and have well-understood cryptographic properties, and they provide good security and performance, whether implemented in hardware or software [2].

Since most 802.11i implementations of AES will be in hardware [3], we intend in this paper to consider different aspects of AES hardware design. Therefore, the rest of the paper is organized as follows: in Section 2 we introduce the CCM protocol and its implications to hardware design; in Section 3 the high performance AES implementation is presented while in Section 4 low cost and in Section 5 low power consumption aspects of the AES implementation are treated. Conclusions are drawn in Section 6.

2. AES-CCM MODE

In CCM mode, the majority of the time of the protocol is spent on computing the AES algorithm. AES is used to generate the cipher text from the header of the 802.11 package as well as the package payload. Therefore, it is incentive to have a hardware assisted ASIC or FPGA for computing the AES cipher text. This will alleviate the computing power from the main processor. One possible implementation of the CCMP protocol is to have the main processor in charge of the MAC layer and ASIC/FPGA device running simultaneously performing the AES encryption/decryption algorithm.

The Advanced Encryption Standard (AES)

Manuscript received July 14, 2004. This work was supported in part by the Spain Ministry of Science and Technology under Grant TIC2003-09061-C03-02 and the "Ramón y Cajal" program.

The authors are with the Electronic Engineering Department, Technical University of Madrid, Spain (e-mails: {slobodan, carreras, jmdiez, nieto}@die.upm.es). The contact person is S. Bojanic.

is likely to become a *de facto* worldwide encryption standard commonly used to protect all means of secret communications during the next several decades. AES (Rijndael) is a symmetric block cipher with a variable key size (128, 192 and 256 bits) and variable input/output block size where only a 128-bits block size is required by the AES Standard. Rijndael is a substitutionlinear transformation cipher based on Sboxes and operations in the Galois fields [4].

The implementation of the encryption round of Rijndael requires the realization of four component operations: ByteSub, ShiftRow, MixColumn, and AddRoundKey, while the implementation of the decryption round requires their inverse operations.

ByteSub includes sixteen identical 8x8 Sboxes working in parallel. Each of these Sboxes can be implemented independently using a 256 x 8bit look-up table. ShiftRow changes the order of bytes within a 16-byte (128-bit) word. This transformation involves only changing the order of signals and, therefore, it can be implemented through routing. The MixColumn transformation can be expressed as a matrix multiplication in the Galois field $GF(2^8)$. AddRoundKey is a bitwise XOR of two 128-bit words and can be implemented using one layer of 128 lookup tables, which translates to FPGA implementation of 64 CLB slices.

The goal of the FPGA is to compute the AES algorithm to generate the cipher text in the range of at least 11Mbps or higher. However, since the 802.11a emergence, it is desirable to have the ASIC/FPGA to run at 54 Mbps. Since CCM spends a lot of computational time on AES decryption/encryption, implementing the AES off chip on a separate FPGA would free the main processor.

Network protocols, particularly IEEE Std. 802.11i, are composed of several different layers. When a user wants to send information over the network, the information is separated into smaller individual packets. These packets are then sent down through the layers and additional information is added to them to ensure proper communication, such as source address and destination address. The packet is then sent through the Ethernet and arrives at the destination where it travels up the layers and is reassembled for the receiver.

One of the layers that packets must go through in 802.11i, is the MAC layer. It is responsible for packing the packet and ensuring proper security. The security scheme that is being proposed is the CCM Mode Encryption that relies heavily on software block ciphers such as AES.

By including the use of a hardware block cipher, the processor can spend some of the computational time on other computations. The ideal goal would be to have this AES hardware implementation embedded on a Wireless Network Adapters for PCs and Laptops. Ultimately this could relieve enough time from the processor to enable faster communication rates and generally both speed up and secure wireless communications.

The AES implementation on the FPGA is a viable solution for improving the speed and processing power of CCM Mode Encryption. The implementation on a Xilinx Spartan II [5] that is partitioned into three modules (input interface, output interface and the AES block cipher engine) shows that a much larger FPGA or ASIC would be preferred, since both encryption and decryption could be implemented as well as some pipelining of processes.

3. HIGH PERFORMANCE IMPLEMENTATION

Most of the AES implementations have been oriented to achieve high performances [6], [7]. In [6] the AES implementation exploits the benefits of the development and application of a pipeline compiler. Several implementations of AES ciphers with keys of 128 bits with sub-key generation in parallel with data processing, have been prepared, analyzed and refined. Figure 1 shows an example of the pipeline compilation process of a combinational block. The conversion flow starts with a fully unrolled version of the algorithm sequence to obtain a data-flow graph. This graph is built considering the time delays of every operator. The next step is to divide the graph in stages by register banks. The compiler decides the positions of these banks and inserts automatically the registers needed.

The AES algorithm can be implemented with three basic operations: XOR of two bit inputs, a tree of XORs which implements the *mixcolumns* function, and a substitution of bytes. The substitution boxes, SBOXs, are tables of 256 bytes whose

The Xilinx devices offer two alternatives to implement memory blocks: distributed memories and RamBlocks. The later option obtains better results for large memories. RamBlocks are specific memory elements of 18Kbits and can be used like single/dual port RAMs or ROMs. The Xilinx tool COREGEN generates directly this block from a bit map supplied by the designer. Figure 2 shows the VirtexII-4000 structure with 6 columns of 20 blocks, the selected manual placement and the flow of data. Every round uses 10 consecutive blocks of one column.

An additional optimization involving the connection of all registers to the reset of the system was also required. The reason for this is an optimization performed by the synthesis tool. The reset connections allow an increment of 15 MHz in the clock frequency, leading to a final frequency of 167.67 MHz and a throughput of 21.5 Gbps. The implementation uses 388 IOBs, 9,357 slices, 100 RAM blocks and 6,734,115 equivalent gates.

A pipelined architecture of AES 128-key without key scheduling is presented in [6]. This design uses 9196 slices and 80 RamBlocks providing a throughput of 16 Gbps versus the 21.5 Gbps obtained here, on a Xilinx Virtex v1000 device. The AES implementation offers excellent results. It proves the possibilities of using a pipeline compiler driving architectural desian. allowing fast design exploration and analysis supporting the of several alternatives and refinements in reduced time.

4. LOW COST IMPLEMENTATION

Much of the research targets state-of-theart technologies where the individual cost of those devices ranges in hundreds of US dollars. These implementations feature high speeds and high costs suitable for high-end applications only [8]. But the need for secure electronic data exchange will become increasingly more important to low-end customer products like wireless devices, thus the AES implementations must become very inexpensive.

Most of the low-end applications do not require high encryption speeds. Current



Figure 1: Example of conversion of a combinational graph (a) into a pipeline architecture (b)

wireless networks achieve speeds up to 60 Mbps. Implementing security protocols, even for those low network speeds, significantly the requirements increases for computational power. For example, the processing power requirements for AES encryption at the speed of 10 Mbps are at the level of 206,3 MIPS [9]. In contrast, a state-of-the-art handset processor is capable of delivering approximately 150 MIPS at 133 MHz, and 235 MIPS at 206 MHz.

Early AES designs were mostlv straightforward implementations of various loop unrolled and pipelined architectures with limited number of architectural which resulted in poor optimisations, resource utilization. For example, AES 8x8 S-boxes were implemented on LUTs as huge tables left for synthesizers to optimise. Later FPGA implementations demonstrated better utilization of FPGA resources. Several architecture usina dedicated on-chip memories implementing S-boxes and Tboxes were developed. Recent research has focused on fast pipelined implementations but most of them are too costly for practical applications [13].

There are few compact implementations of the AES algorithm in FPGAs. There exist



Figure 2: Manual placement of RAM-Blocks

commercial compact cores from Amhion [10] and Helion [11] companies. Both companies provide compact cores in encryption or decryption version only, and a 128-bit key schedule.

An analytical approach to compact the AES implementation resulted in the proposal of an AES S-box implementation based on composite fields [12]. Another approach to create a low-cost implementation of AES in the FPGA targeted for real life applications is to shift attention to older technologies and smaller devices. The implementation [13] fits in an inexpensive, off-the-shelf Xilinx Spartan II XC2S30 FPGA, with a cost that starts below \$10 per unit. Only 50% of the logic resources available in this device were utilized, leaving enough area for additional logic. This implementation can encrypt and decrypt data streams up to 166 Mbps. The

encrypted speed, functionality, and cost make this solution perfectly practical in the world of embedded systems and wireless communication.

5. LOW POWER CONSUMPTION

The IEEE 802.15.4 Low-Rate Wireless Personal Area Network (LR-WPAN) standard [14] also requires the use of the AES algorithm in CCM mode for wireless sensor networks where low power consumption is the priority. The networks should operate for many months from relatively small batteries, in sensing applications that range from industrial control and monitoring, security, and home automation to those found in health care, telematics, and intelligent agriculture. To attain low-power operation, it is necessary to minimize the device's power consumption while it is active for both the sensor (or actuator) and the communication transceiver. The sensor use is application specific, and its consumption can exceed that of the transceiver. Also the power consumed during standby mode can easily become the dominant component in the average power consumption calculationbecoming more significant than the power consumed during active operation.

The transmission of the unsecured packet with a first-generation transceiver requires $56 \mu J$ assuming that the data payload is 16 bytes, that short (logical) addresses are used in the medium access control (MAC) header, and that the 2.4 GHz band is employed (which has a data rate of 250 Kbps) [15]. When CCM is applied, 13 additional bytes must be transmitted in the packet (8 for the MIC, 4 for the frame counter field, and 1 for the key sequence counter), thus transmission of the secured packet requires an additional 21 μJ .

If the AES security calculations are performed in software by a microcontroller, e.g., the Motorola MC9S08GB60, from its data sheet [16], the energy needed to perform the calculations is then: $35 \ \mu$ J. So $56 \ \mu$ J are needed to transmit this packet in an unsecured mode, while an additional 21 + $35 = 56 \ \mu$ J are needed to transmit this packet in the CCM secure mode, using software to encrypt it. Doubling the energy requirement to incorporate security may seem onerous, but recall that encryption functions are only performed when packets are transmitted or received and that IEEE 802.15.4 is by definition a low-data-rate network with relatively low throughput. Security functions are therefore rarely performed. As noted earlier, in these systems standby power consumption often dominates active power consumption in the battery life calculation.

Alternatively, a dedicated AES engine may be implemented in hardware to reduce consumption further. power The performance parameters of such a device are highly implementation specific; however, we can still make rough conservative estimates [15]. If we assume the engine draws 1 mA from 2 V and requires 100 cycles at 16 MHz to produce an output, the security calculation will require 87.5 nJ. This is significantly less power than that required by the software approach, but it comes at the cost of a dedicated circuit that must be purchased and there is still the need of 21 µJ to transmit the additional 13 bytes in the secure packet.

6. CONCLUSION

As 802.11i becomes a standard in the near future, it will replace the current 802.11 WEP security scheme and the AES algorithm in CCM mode will be the basis for data protection across the wireless medium. Since most 802.11i implementations of AES will be in hardware, there is strong need of thorough examination of each of the components of the AES algorithm for a perfect match into the architecture of the FPGA or ASIC. This paper considers the issues of high performance, low cost and low power design issues in the AES implementation and demonstrates obtained results as well as further research directions.

REFERENCES

- R. Housley, Panel Discussion on the Differences and Similarities of Wired vs. Wireless Security, OpenSig 2003, 9 October 2003.
- [2] D. Whiting, R. Housley, and N. Ferguson: IEEE P802.11 wireless LANs: AES encryption & authentication using CTR mode & CBC-MAC. IEEE Tech. Rep. IEEE 802.11-02/001r2, May 2002.
- [3] N. Ferguson: AES Mode Choices: OCB vs. Counter Mode with CBC-MAC. IEEE 802.11-01/634r0, November 2001.
- [4] U.S. Dept. of Commerce, National Institute of Standards and Technology, Information Technology Laboratory, Specification for the Advanced Encryption Standard (AES). Federal Information Processing Standard Publication (FIPS PUB) 197. Springfield, VA: National Technical Information Service. 26 Nov. 2001.
- [5] K. Vu, D. Zier: FPGA Implementation AES for CCM mode encryption using Xilinx Spartan II. ECE-679, Oregon State University, Spring 2003.
- [6] J. M. Díez, Š. Bojanic, C. Carreras, O. Nieto-Taladriz: FPGA Implementation of Three IPSec Cryptographic Algorithms. WSEAS Trans. on Systems, Issue 1, Vol. 2, p. 229-234, 2003.
- [7] A. Hodjat and I. Verbauwhede: A 21.54 Gbits/s fully pipelined AES processor on FPGA, IEEE Symposium on Field-Programmable Custom Computing Machines, April 2004.
- [8] P. Chodowiec, K. Gaj, P. Bellows, and B. Schott, Experimental Testing of the Gigabit IPSec-Compliant Implementations of Rijndael and Triple DES Using SLAAC1V FPGA Accelerator Board, Proc. Information Security Conference, Malaga, Spain, October 2001.
- [9] Ravi S., Raghunathan A., Potlapally N.: Securing Wireless Data: System Architecture Challenges, Symposium on System Synthesis, 2002.
- [10] Amphion: http://www.amphion.com.
- [11] Helion: http://www. helion.com.
- [12] Rijmen V.: Efficient implementation of the Rijndael S-box, available at: http://www.esat.kuleuvan.ac.be/rijmen/rijndael/sbo x.pdf.
- [13] P. Chodowiec, K. Gaj: Very compact FPGA implementation of the AES algorithm. LNCS 2779, pp. 319-333, 2003.
- [14] Institute of Electrical and Electronics Engineers, Inc., IEEE Std. 802.15.4-2003, New York: IEEE Press. 2003.
- [15] E. Callaway: Secure Low-Power Operation of Wireless Sensor Networks, Sensors, http://www.sensorsmag.com/articles/0104/22/main .shtml, January 2004.
- [16] Data sheet for Motorola MC9S08GB60 microcontroller.