Streaming Encryption for a Secure Wavelength and Time Domain Hopped Optical Network

Herwin Chan, Alireza Hodjat, Jun Shi, Richard Wesel, Ingrid Verbauwhede
{herwin, ahodjat, junshi, wesel, ingrid} @ ee.ucla.edu
Electrical Engineering Department
University of California, Los Angeles

Abstract

This paper describes a working implementation of a streaming encryption system for optical networks. The 10 Gbps data stream is encrypted on the physical level in both the wavelength and time domains. Security is obtained by applying a strong pseudo-random hopping pattern to both. The AES algorithm in counter mode is used to control the switches that implement the hopping permutations. Because of the high throughput requirements, aggressive parallelizing and pipelining techniques are used to achieve data rates of 10 Gbps.

The core logic of the system was implemented on the Virtex2-XC2V1000 FPGA. Using four FPGA boards and four serializer and deserializer chips, a secure switch of 4 users over 4 wavelengths can be realized, resulting in a total throughput of 10 Gbps. This system is important as a platform for further research in the area of secure optical networks.

1. Introduction

Physical layer security is one way of increasing the security of an entire network. In such a system, all data from a source is encrypted before being put onto the network. This paper describes an electronic implementation of a streaming encryption system for optical networks. The basic concept involves the scrambling of the incoming bits of a datastream in both the frequency and time domains. High throughputs can be achieved using aggressive parallelization and pipelining techniques. Previous work in this area has focused on design of optical components which will perform the coding in the optical domain [1, 2]. Such systems have not yet been realized due to design complexity of the multi-wavelength hardware components.

Other systems focus on development of streaming encryption coprocessors [3, 4]. Because such approaches do not take a system view, interfacing overheads can severely degrade performance throughput. The design of our system incurs minimal overhead and can therefore achieve much higher throughput demands.

The main contribution of this paper is a description of a functional and scalable streaming encryption system. In our system, the cryptographically strong encryption algorithm, AES, is used in the counter mode to generate a pseudo-random bit stream. This bit stream is used as control signals which directly permute the data stream through a switch matrix. By implementing the complex logic of the code generator outside the datapath, high rates can be achieved. A further benefit of this architecture is that the level of security can be easily scaled by varying the rate of the code generator.

Section 2 describes the main coding strategy used in the security system and defines the CDMA concept of the optical links. Section 3 then describes the overall architecture of this system and how different units of this system were implemented using the FPGA platform. Section 4 presents the complete system integration of the proposed security scheme. Section 5 discusses the performance and cost of such a system and is followed by the conclusion in section 6.

2. Optical CMDA concept

Code Division Multiple Access, (CDMA), has been recognized to provide efficiency, security, and multi-access benefits in wireless communications. This has triggered interest in providing similar advantages for optical communication systems. Our encryption system takes the basic ideas of CDMA and implements it on top of the traditional Wave Division Multiplexing (WDM) system used in optics.

In traditional WDM, the input data stream is demultiplexed to be transmitted on a number of wavelengths (Figure 1a). If each wavelength represents the data of an individual user, an attacker need only look at this one wavelength to understand the information.

In our proposed coding scheme, this data stream is broken up into frames of N bits by M wavelengths (Figure 1b). Each frame contains NxM elements representing a unique time-wavelength position. The coding scheme will make a random permutation of these elements for each time frame. This means both the order of the bits in a stream and the wavelength in which it is transmitted may change.

Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC'04)
0-7695-2108-8/04 $ 20.00 © 2004 IEEE

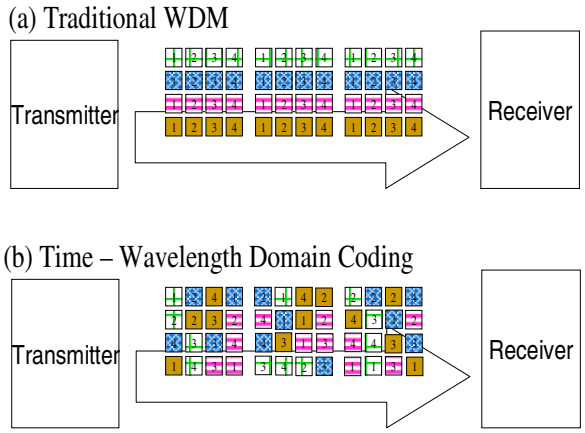

Figure 1. Time-wavelength coding compared to traditional WDM transmission scheme

The security of the system depends on the quality of the random permutations. A strong pseudo-random number generator, based on the AES algorithm in counter mode is chosen to control the hopping patterns. Important for this is that the key is kept secret and that the initial seed is non-repeating.

3. Implementation

This section presents the overall architecture and the implementation of the proposed security scheme. The FPGA devices are used to implement the hardware units of the overall architecture.

3.1. System architecture

The demonstration setup encrypts and decrypts a data stream at a data rate of 10 Gbits/s. Data is transmitted through a single fiber optic cable carrying signals modulated with four separate wavelengths. The width of the data frame is 4 bits. This means that 16! unique permutations are possible for each frame.

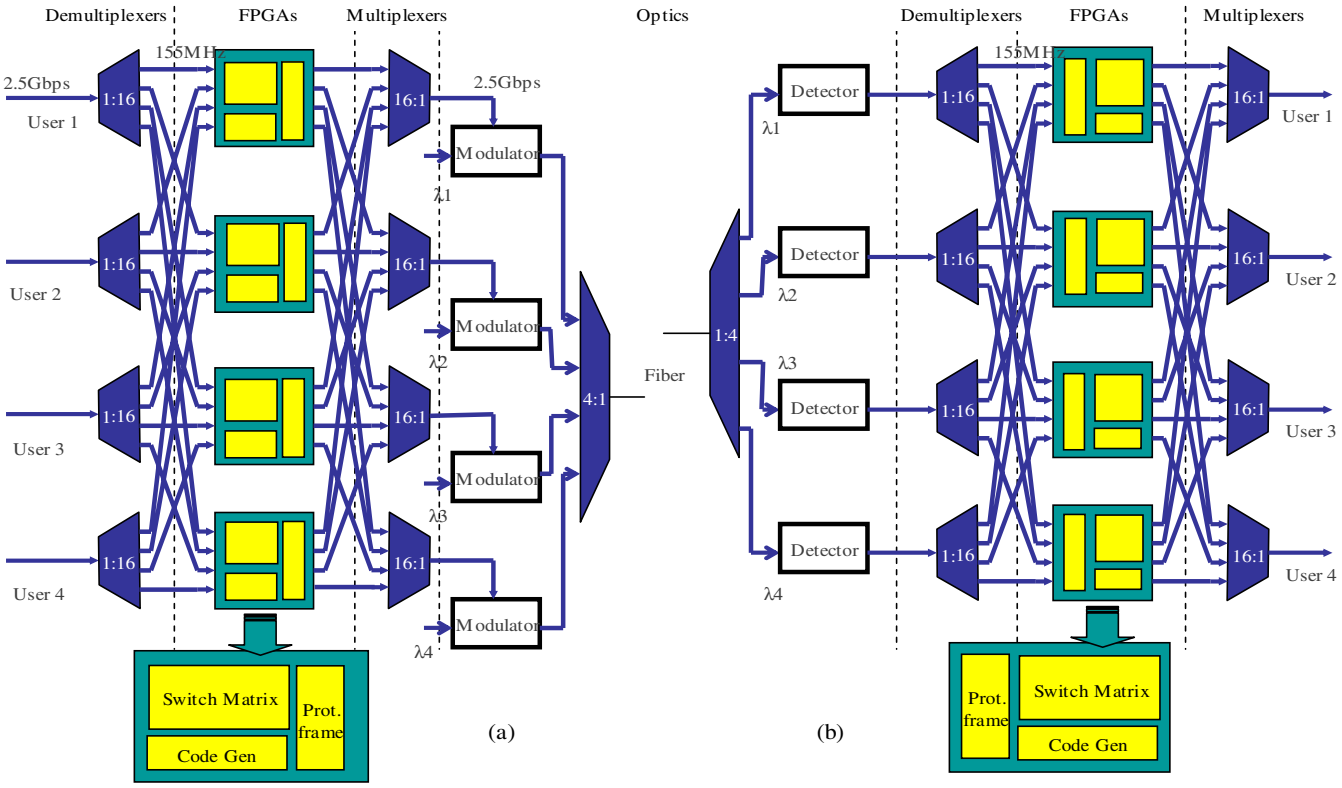

Figure 2. Schematic view of (a) encryption and (b) decryption systems

Figure 2a shows a high level description of the encryption system. Four high speed data lines (2.5 Gbps) enter into the encryption block. They are de-serialized to 155 Mbps so that the processing can be performed in the digital domain. A series of 16x16 switches controlled by a code generator is used to produce a random permutation of the inputs. Outputs from the switches are then serialized and modulated before being transported out through a fiber connection.

The high level description of the decryption system is shown in Figure 2b. On the decryption side, the opposite operations must be done. However, it must be ensured that the code generators produce a synchronized and corresponding code in order for the original data streams to be recovered.

Serialization and de-serialization is performed by discrete electronic components. Its main role is to parallelize the input data stream to a manageable rate.

FPGA's are used to implement the switching functions and provide the pseudo-random code generation and synchronization. Modulation and demodulation of the data signals are performed by off the shelf optical components.

3.2. FPGA implementation

The Virtex2-XC2V1000 FPGA from Xilinx was used as the platform for the encryption and decryption units. This provides us with the necessary speed while being programmable so that different architectures and features can be added to the system easily.

Figure 3 shows the overall architecture of the digital encryption unit implemented on the FPGA. It has the following main components: switching matrix, code generator unit, and protocol unit. The following subsections describe the details of each unit in this security system as well as the token passing technique which synchronizes them.

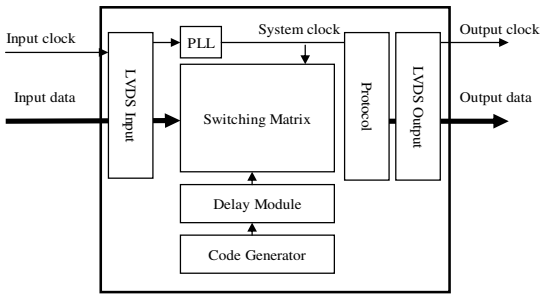

Figure 3. Functional blocks in FPGA encryption

3.2.1. Switch matrix. The switching matrix is composed of a network of 2x2 switches which allows the output to be any permutation of the input. Figure 4 shows how the 56 switches are connected in a Benes network [5]. Each of these switches is controlled by a signal coming from the code generator. New control bits are produced each clock cycle.

The basic building block of the switch matrix is the 2x2 switch. A register is present at each output of this switch to ensure that the FPGA can be used at a high clock rate. This pipelining allows for high throughput with the side affect of increased latency.

Because data path pipelining is introduced by putting registers in the switches, the control signals for the switches must also be appropriately delayed. Figure 4 shows how such a scheme should be implemented during encryption.

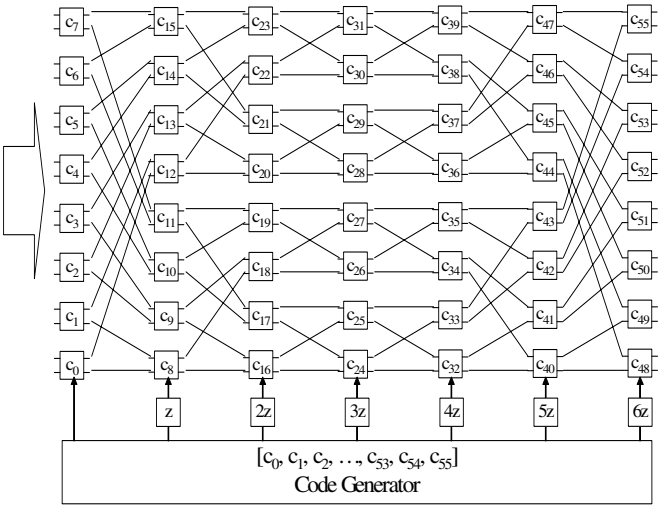

Figure 4. Switch matrix configuration for encryption

During decryption, the control signals must be flipped so that it looks like a mirror image of the encryption structure. This is so that the code signals would match properly and the original data streams can be recovered at the output.

3.2.2. Code generator. The code generator unit is responsible for generating the pseudo-random numbers which will act as control signals to the switches in the switching matrix. The following two approaches are considered for this purpose.

3.2.2.1. AES in the counter mode of operation. In order to generate cryptographically secure random numbers, the Advanced Encryption Standard [6] can be used. Figure 5 shows how the AES algorithm can be

implemented in the counter mode of operation [7] for random number generation. The initial 128-bit seed is loaded into the seed register. This seed forms the initial value of the counter register. Every clock cycle the counter value is incremented. The counter value is loaded into the AES unit and it is encrypted. The encrypted value is the generated random number that is written out. Starting form a secure non-repeating initial seed, 2^{128} sequences of 128-bit random numbers are generated. Moreover, the throughput rate of the random number generation is equal to maximum throughput of the AES algorithm.

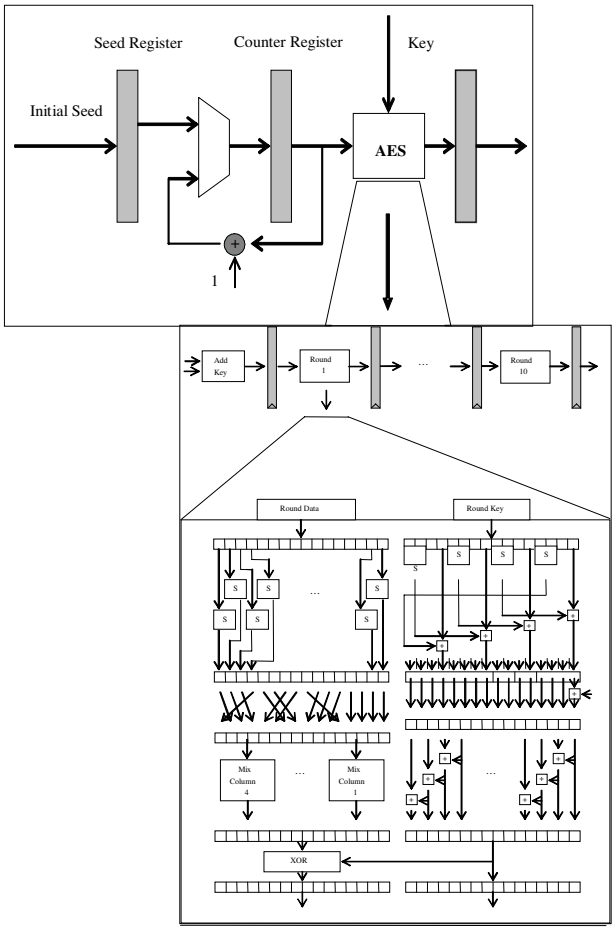

Figure 5. High throughput AES in the counter mode of operation

Figure 5 shows the architecture of the AES algorithm that can achieve a throughput rate of an over 30 Gbps. This is fully pipelined implementation of the Advanced Encryption Standard. The key length is limited to 128 bits. This helps to reduce the critical path of the key scheduling datapath. The encryption algorithm is loop-unrolled and pipeline registers are defined between each round of the algorithm. Since all the 10 rounds of the algorithm are unrolled and pipelined, one output sample is generated every clock cycle. Therefore, the throughput rate is equal to 128 bits times the clock frequency of the design. References [8] and [9] present different pipelined architectures and the throughput-area trade-offs of several ten Gbps AES processors for both FPGA and ASIC implementations. The architecture of figure 5 is one of the simplest proposed designs of [8] and [9] which can achieve the throughput rate of over 30 Gbps in the 0.18 μm CMOS technology.

3.2.2.2. Linear feedback shift registers. Though it is not cryptographically secure, the code generator is currently tested with a 56 bit linear feedback shift register (LFSR). The output value of each of these registers will be used as the control signal.

In the future, this implementation will be replaced the by a secure implementation of the AES algorithm as explained in the above section. The demonstrated 30 Gbps achievable in CMOS technology suggests that an FPGA implementation can meet our system requirements of 9 Gbps per channel.

3.2.3. Token passing. In a high throughput security system as the one proposed in this paper, it is vital that proper synchronization is established and maintained. This means that the code generators at both the transmitter and receiver are activated at the proper time to decode the data stream. The method used in this system for synchronization is token passing.

The token is an extra input into the system and exists when data is to be encrypted. It will travel through the system together with its associated data and experience the same latencies in each of the blocks.

Transmission of the token through the optical channel is accomplished in the protocol block. This block adds token information into the data stream by inserting code words.

3.2.4. Protocol. From the output of the switches, the signals can be thought of as a stream of 16 bit words. The output data stream from the encrypt switch is divided into blocks of 255 words. Before transmission, a synchronization word is inserted between each of the blocks before transmission. The synchronization word can be one of two predetermined values and indicate the presence or absence of a token in the proceeding block of data.

The implementation of this functionality involves the creation of two extra blocks of hardware: the SYN symbol inserter (for the transmitter) and the SYN symbol detector / remover (for the receiver).

4. Complete system integration

Figure 6 shows the architectural diagram of the whole system. It includes all the elements that were shown in Figures 2 and 3. There is a total of 8 FPGA's in the complete system. The serializer and deserializer units on both the transmitter and receiver sides are integrated onto a single board, indicated by SERDES. On the transmitter side the encrypted output of each serializer goes to the optical modulator before combination to a single optical fiber through the optical multiplexer. On the receiver side, optical data is first demultiplexed and demodulated before being converted back to an electrical signal.

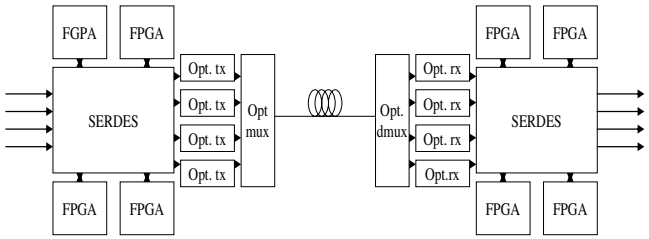

Figure 6. Component architecture of demonstration system

5. Results

Serialization and deserialization of the four 2.5Gbps data streams were accomplished with commercial off the shelf SERDES components. It communicates with the Xillinx FPGA, which performs the actual encryption/decryption, through an LVDS interface.

Table 1 shows the implementation cost of each block implemented in the FPGA. Note that though the implementation costs are very low in such an implementation, much of the area in the future will be taken up by our secure AES code generator. In addition, system size is expected to shrink in the future as the SERDES functions are integrated onto the FPGA.

Table 1. Implementation cost of encryption/decryption

Block	Area (slices)	Speed (ns)
Switching matrix	155	2.1
Protocol (encryption)	54	3.3
Protocol (decryption)	29	4.9
Code Generator (LFSR)	29	2.5

The circuit speed is within the requirements needed for 10 Gbps operation and can currently sustain rates of up to 13 Gbps. Futher optimizations of the protocol blocks can increase this throughput even more.

6. Conclusion

A working implementation of a secure streaming encryption system is described in this paper. High throughputs are achieved by using a code generator to directly control the datapath of the datastream. Aggressive pipelining and parallelization were used to reduce clock speeds to a manageable level.

Because of its small size and flexible implementation platform, it is a good platform in which to explore the properties of such systems. Future work will focus on component consolidation and studies on system scalability. As a research platform, it can be used to study higher level issues such as protocol development and fault tolerant architectures.

Acknowledgement

This material is based upon work supported by the Space and Naval Warfare Systems Center - San Diego under contract No.N66001-02-1-8938. The authors would like to acknowledge the funding of this project.

References

[1] S. Yegnanarayanan, A. S. Bhushan, and B. Jalali, "Fast wavelength-hopping time-spreading encoding/decoding for optical CDMA", IEEE Photonics Technology Letters, vol.12, (no.5), IEEE, May 2000.

[2] C. F. Lam and E. Yablonovitch, "A fast wavelength hopped CDMA system for secure optical communications", Proceedings of the SPIE - Dallas, TX, SPIE-Int. Soc. Opt. Eng, 1998.

[3] D. Carlson et al, "A high performance SSL IPSEC protocol aware security processor", ISSCC 2003.

[4] D. K. Y. Tong et al, "A system level implementation of Rijndael on a memory-slot based FPGA card", IEEE Conference on Field-Programmable Technology, December 2002.

[5] C. Chang and R. Melhem, "Arbitrary Size Benes Networks", Parallel Processing Letters, vol 7, no 3, pp 279-284 (1997).

[6] National Institute of Standards and Technology (U.S.), Advanced Encryption Standard. Available at: http://csrc.nist.gov/publication/drafts/dfips-AES.pdf

[7] M. Dworkin, SP 800-38A 2001, "Recommendation for Block Cipher Modes of Operations", December 01.

[8] A. Hodjat, I. Verbauwhede, "Speed-Area Trade-off for 10 to 100 Gbits/s throughput AES processor", 37th Asilomar Conference on Signals, Systems, and Computers, November 2003.

[9] A. Hodjat, I. Verbauwhede, "Minimum Area Cost for a 30 to 70 Gbits/s AES Processor", IEEE Computer Society Annual Symposium on VLSI, February 2004.