Evaluation system of dynamically changing cryptographic algorithms using the SEBSW-1:PCI-based encryption and decryption PC board

Hirotsugu KAJISAKI and Takakazu KUROKAWA
Department of Computer Science , National Defense Academy
1-10-20 Hashirimizu, Yokosuka-shi, 239-8686 Japan
TEL +81-468-41-3810(ex.2219) FAX +81-468-44-5911
E-mail: g40073@nda.ac.jp, kuro@nda.ac.jp

Abstract: In a network communication process, cryptographic algorithms play important role for secure process. This paper presents a new system architecture named "DCCS." This system can handle flexible operations of both cryptographic algorithms and the keys. For experimental evaluation, two representative cryptographic algorithms DES and Triple-DES are designed and implemented into an FPGA chip on the SEBSW-1. Then the developed board is confirmed to change its cryptographic algorithms dynamically. Also its throughput confirmed the ability of the real-time network use of the designed system.

1. Introduction

Cryptographic algorithms are playing important role to keep secure network communication process. As the traffic of the Internet increases every year, high speed process for encryption as well as decryption requires the specific devices such as PLD and ASIC. Following these circumstances, we focused on a system that changes not only the keys but also the cryptographic algorithms dynamically.

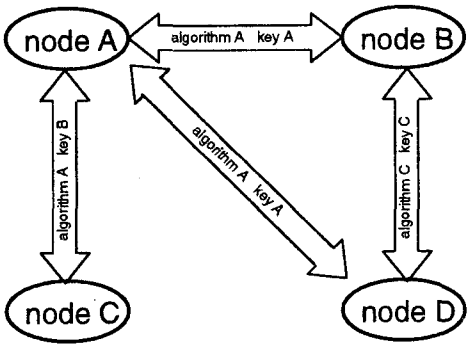
Several approaches have tried to keep secure communication among each node(i.e. IPsec [1],TLS [2],SSH [3],etc.). Most of them are implemented in the application layer. Thus, when we use several applications as reading E-mail, browsing the WWW, and connecting other PCs, we must manage different cryptographic algorithms and also different keys according to these applications.

In this paper, we will propose a new system which changes both cryptographic algorithms and the keys arranged in the same layer for the simplified security management with high performance. Furthermore the dynamic change of cryptographic algorithms prevents the conventional cipher attacks. Then a design and FPGA based implementation of SEBSW-1 (SEcret-key Block cipher SWitcher) board will be presented. This board is designed as an encryption and decryption board to accord with the dynamic change of cryptographic algorithms. As a cryptographic library on SEBSW-1, typical block ciphers (DES and Triple-DES) are designed and implemented into this board, and their performance evaluation with will be also discussed.

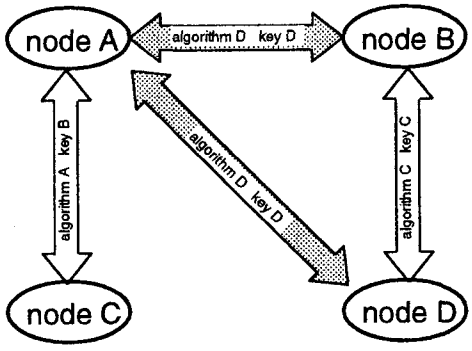
2. DCCS:Dynamically Changeable Ciphers System

For a secure transformation of massive data, a system must keep high security level without decreasing its secure level against several kinds of security attacks. If a system can manage the cryptographic algorithms and the keys at the same layer, a simplified security management system can be established. Additionally, FPGA has the flexibility to be application specific circuits with high computational power. Thus FPGA is suited to process the cryptographic algorithms.

Concerning to these backgrounds, we present a Dynamically Changeable Ciphers System (DCCS) that changes not only keys but also cryptographic algorithms dynamically using reconfigurable devices such as PLDs and FPGAs. The overview of such a system, which consists of 4 nodes as an examples, is shown in Figure 1. Each of the four communication nodes is able to link with the other nodes using the encrypted data.



(a) Initial condition of four nodes.



(b) Next condition after changing algorithms and keys used for communication among nodes A, B and D.

Figure 1. Example of DCCS.

For example, the initial condition of this system is shown in Figure 1(a). The node A is able to communicate with node B and D using algorithm A and key A. This node A is also able to communicate with node C using algorithm A and key B. Let us think about the next condition after changing algorithms and keys to be used for the communication between node A and node B, D as shown in Figure 1(b). The algorithm and key used for the communication among these nodes is changed from A to D.

3. Evaluation system

3.1 Overview

Following the above presented architecture, we are developing an evaluation model using IBM PC/AT and PC-UNIX as shown in Figure 2. The two SEBSW-1 boards presented at the right side in this figure have encryption and decryption circuits using FPGAs. These boards are equipped with different interface circuits (PCI and USB) for the communication with the host PC. The storage presented at the left side in this figure is the Hard Disk Drive that contains configuration data of FPGAs in SEBSW-1 boards as a cryptographic library.

3.2 SEBSW-1:PCI-based encryption and decryption PC board

We developed the SEBSW-1 as an encryption and decryption board with the PCI interface. The block diagram and the implemented board of SEBSW-1 are shown in Figure 4 and Figure 3 respectively. The SEBSW-1 contains a FPGA as encryption and decryption circuit, a PCI controller for interfacing PCI bus, a bus controller to control the local bus, a SRAM for storing configuration data, and an IO controller to assert control signal. The configuration data of FPGA that accords with each cryptographic algorithm is stored in the SRAM. When we use an algorithm 'A', its configuration data is loaded to the FPGA from the SRAM. Then the configuration signal from host PC through the PCI controller is asserted. After configuration, DONE signal is asserted by the FPGA, and the algorithm 'A' is implemented in the FPGA circuit.

The details of components of SEBSW-1 are summarized in Table 1.

Table 1. Type and package of components in SEBSW-1.

device	company	type	package
IO	NEC	μPD71055C-10	DIP40
SRAM	Hitachi	HM628512ALFP-7	SOP32
BUS Controller	Xilinx	XC95108	QFJ84
FPGA	Xilinx	XCV300PQ240-4	QFP240
PCI controller	Zenic	ZEN7201AF	QFP144

3.2.1 PCI controller

Because of experimental implementation, we selected the PCI controller in SEBSW-1 considering the simple functionality. Zenic ZEN7201AF PCI contorller can follow 33 MHz and 32-bit operation for PCI bus, and 16-bit and 33 MHz maximum clock speed for local bus. In this experimetal implementation, the local bus of SEBSW-1 consists of 8-bit and work with 16 MHz cycles.

3.2.2 FPGA configuraion

The FPGA chips of Xilinx Virtex series has the following four configuraion modes;

- Slave-serial mode,
- Master-serial mode,
- SelectMAP mode, and
- Boundary-scan mode.

In this experimental implementation, it needs that the context switching should be as fast as possible. Followingly, the configuration mode of SEBSW-1 is decided to use the selectMAP mode.

3.2.3 Local bus controller

Local bus controller is implented by two CPLDs (Xilinx XC95108-7). It outputs two kinds of contorol signals. One is a control signal of data and address bus. The other is a configuraion control signal for FPGA.

4. Basic performance of SEBSW-1

The basic performance of SEBSW-1 is shown in Table 2. The first item is the bandwidth of the communication between the host PC and SEBSW-1. The read cycle that the data transform from SRAM on SEBSW-1 to main memory in host PC is approximately 16 MB/sec. The write cycle that the data transform from main memory in host PC to SRAM on SEBSW-1 is approximately 10 MB/sec. These bandwidths are measured by sending or receiving the 1-500KB data between the host PC and SRAM on SEBSW-1. The second item is the configuration time of FPGA from asserting the configuration signal by the IO controller till the assertion of the DONE signal by FPGA.

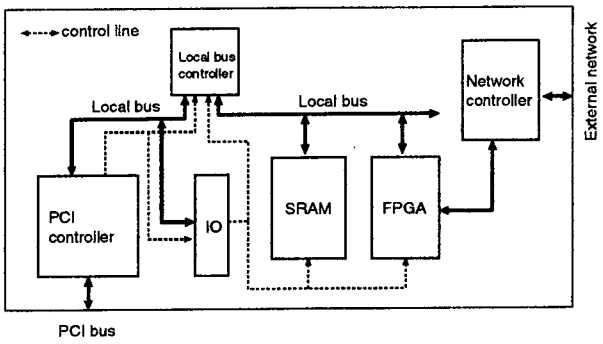


Figure 4. Block diagram of SEBSW-1.

Table 2. Basic performance of SEBSW-1.

	direction	
bandwidth	Read cycle	10 [MB/sec]
	Write cycle	16 [MB/sec]
configuration speed		15.323 [msec]

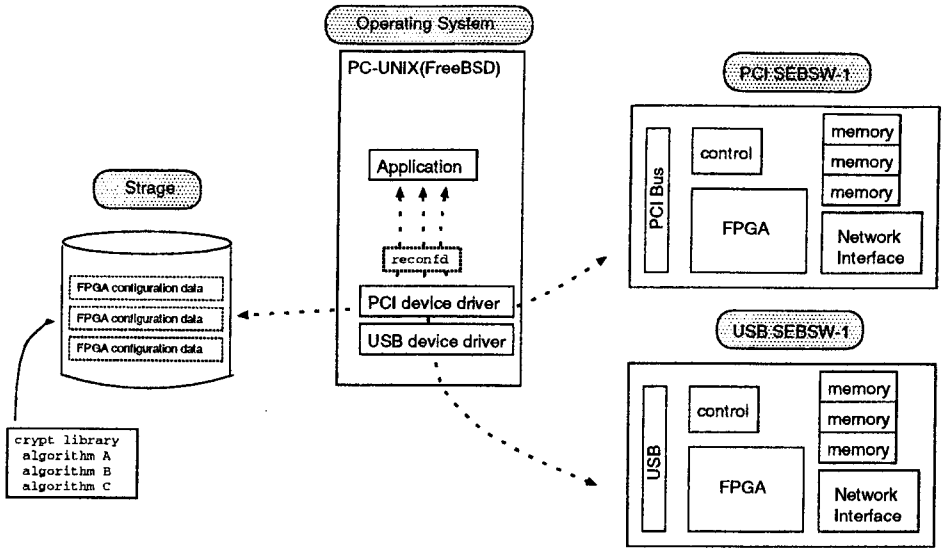


Figure 2. Structure of each communication node.

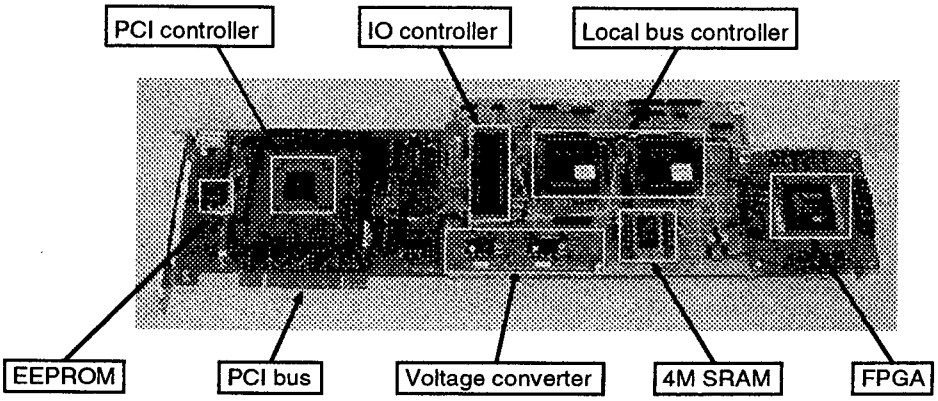


Figure 3. Front view of SEBSW-1.

5. Implementation of block ciphers

DES and Triple-DES are typical Feistel type block ciphers. There are several research groups to implement DES and Triple-DES on FPGAs [4–7]. Their approaches tend to implement as fast as possible using the large scale devices like Xilinx Virtex XCV1000. Using such a large device has advantages for high performance. However, the configuration time tends to increase due to the size of configuration data which depends on the device scale. Consequently, the optimization between the target device and the performance of the implemented cipher must be required.

5.1 Loop architecture

The representative architectures to implement the secret-key block ciphers are the fully loop unrolled architecture, the Loop architecture, and the pipeline architecture. The characteristic point of the loop architecture is sufficient to implement only one round function, so it has the advantage for implementation area. Consequently, we have decided to implement the loop architecture for SEBSW-1. The general loop architecture is shown in Figure 5. The multiplexer presented as "mux" selects new block data or feedback output of round. The length of the input/output data and the key, and the number of round of DES and Triple-DES are summarized in Table 3.

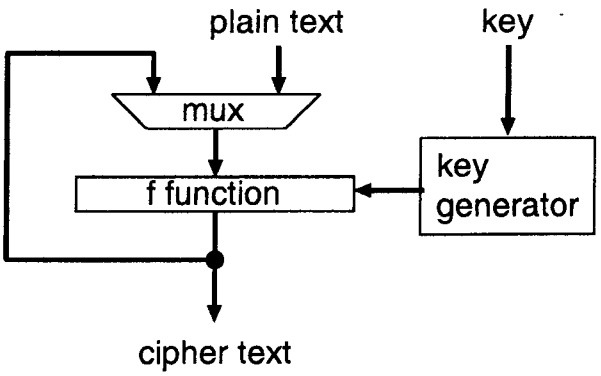


Figure 5. Loop architecture.

Table 3. Bandwidth of data, key, and the number of round.

Type	In/Out[bit]	key[bit]	Round
DES	64	56	16
Triple-DES	64	112	48

Table 4. Implementation results of DES and Triple-DES.

Type	slices	Frequency	Throughput
DES	589	31.42 MHz	125.67 Mbps
Triple-DES	656	35.91 MHz	47.89 Mbps

5.2 Developing environment

We used Xilinx Foundation ISE4.1 for simulation, synthesizing, translation, mapping, place and routing. Loop architecture of DES and Triple-DES were implemented at RTL level in VHDL.

6. Performance evaluation

Each design was synthesized and placed and routed on the target device(Virtex XCV300PG240-4) with I/O registers. The implementation results of DES and Triple-DES are summarized in Table 4. The latency of each design equals to the number of rounds shown in Table 3. Concerning the throughput, DES is approximately tree times higher than Triple-DES. This is the cause for the difference of the latency.

As a result, it is confirmed that the throughput of both designs could be over 10 Mbps. Then, these designs can be used by connecting with the network controller like an Ethernet. There are still remaining resources, so that further speedup will be available if we select other architecture of cipher such as the pipeline architecture for the hardware implementation.

7. Conclusion

In this paper, we proposed the Dynamically Changing Ciphers System (DCCS) to establish secure data link by changing not only key but also cryptographic algorithms dynamically. An experimental implementation of SEBSW-1(PCI-based encryption and decryption PC board) and its basic performance evaluation are also shown.

Further evaluation of the extended architecture of SEBSW-1 including the network interface circuits will be the future work. Moreover, we are still designing and evaluating other secret-key block ciphers, using SEBSW-1 to be included as parts of cryptographic library.

References

[1] S. Kent and R. Atlinson: "IP Encapsulating Security Payload (ESP)", RFC 2406 (1998).

[2] T. Dierks and C. Allen: "The TLS protocol version 1.0", RFC 2246 (1999).

[3] T. Ylonen: "SSH—secure login connections over the Internet", Proc. of the Sixth USENIX Security Symposium, pp. 37–42 (1996).

[4] K. Gaj and P. Chodowiec: "Comparison of the hardware performance of the AES candidates using reconfigurable hardware", The Third Advanced Encryption Standard Candidate Conference (Ed. by NIST), National Institute for Standards and Technology, pp. 40–56 (2000).

[5] P. Chodowiec, K. Gaj, P. Bellows and B. Schott: "Experimental Testing of the Gigabit IPSec-compliant Implementations of Rijndael and Triple DES Using SLAAC-1V FPGA Accelerator Board", Proc. Information Security Conference (2001).

[6] S. Trimberger, R. Pang and A. Singh: "A 12 Gbps DES encryptor/decryptor core in an FPGA", Workshop on Cryptographic Hardware and Embedded Systems(CHES2000), pp. 156–163 (2000).

[7] P. Chodowiec, P. Khuon and K. Gaj: "Fast implementations of secret-key block ciphers using mixed inner- and outer-round pipelining", Proc. ACM/SIGDA Ninth International Symposium on Field Programmable Gate Arrays, FPGA'01, pp. 94–102 (2001).