

IC-ICTES 2011, Pattaya, Thailand 27-29 January 2011

FPGA Implementation of FDE-Portable Hard Disk System

Chalermwat Thanavijitpun

College of Data Storage Technology and Applications
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
chalermwat.thanavijitpun@seagate.com

Khanob Thongkhome

College of Data Storage Technology and Applications
King Mongkut's Institute of Technology Ladkrabang

Bangkok 10520, Thailand
raksy.thongkhome@gmail.com

Somsak Choomchuay

Dept. of Electronic Engineering, Faculty of Engineering
King Mongkut's Institute of Technology Ladkrabang
Bangkok 10520, Thailand
kchsomsa@kmitl.ac.th

Abstract—This Regarding the well growing of data security and privacy, the total-secured data storages are drawing more interest. This paper presents the implementation of FDE portable hard drive that includes AES in its data encryption/decryption routines. Rather than using ordinary password in the authentication process, our design incorporates a technique of fingerprint authentication. It is simple and user friendly but offers good security. The FDE hardware is basically inserted between the USB port and an ATA hard drive. It complies with USB 2.0 data rate (480 Mbit/sec). ATA protocol command decoder and AES core are implemented with FPGA whilst USB interface controller and fingerprint scanner are supported by custom chips. The obtained simulation result has confirmed us the functionality of the designed system.

Keywords- Full Disk Encryption, FDE, AES, FPGA

I. INTRODUCTION

As data privacy issue is now more aware, the importance of organization of confidential data stored on hard drives in PCs and/or servers cannot be no more ignored. Unfortunately, only few portable hard disk drives nowadays do include data security system/function to protect information that may be critical for individuals and mandatory for business. There could be various security threats, such as the malicious data modification, data leaks, and lost hard-disk. Events may cause inestimable loss to some organizations, such as military, governments, and enterprises. Regarding the secured data importance, several risk protection schemes have been implemented. Those may vary in terms of security strength, user friendliness, and cost of implementation.

BIOS and operating system passwords are commonly used but these efforts can offer very limited security. They can be easily removed even by the un-expert attackers. Hard drive protection password is more difficult to crack or remove but the security strength is still not so strong. Full disk encryption (FDE or whole disk encryption) uses disk encryption software or hardware to encrypt every bit of data that goes through a disk or disk volume. The FDE is significantly stronger than the first two methods mentioned above. The security strength depends mainly on the strength of the cryptographic algorithm used.

The software-based FDE method uses the computer's CPU to perform encryption/decryption tasks. This method has shown some disadvantages: (1) The encryption/decryption software can be easily monitored by a Trojan program. (2) The instructions of the encryption/decryption are executed by the CPU. Obviously the operations consume more computer resources. (3) It is difficult to transfer the encryption/decryption software among different operating systems.

The hardware-based method uses specific designed chips to accelerate the encryption/decryption process. Despite the hardware complexity, this method needs less computer resources. The security of hardware-based hard disk encryption/decryption is higher than that of the software-based hard disk encryption/decryption.

Advance Encryption System (AES) [1] accepted by NIST as FIP-197 in the year 2001 is a popular algorithm used in existing FDE systems. In both of software-based and the hardware-based encryption/decryption system, the key is basically stored in special sectors of the hard disk. Although there is a rumor around that AES can be cracked, there is no clear evident or no confirmed report.

In this paper, we partially outline the hardware-base FDE implementation. According to the implementability and the corresponding feasibility validation, the design is targeted at FPGA realization. Despite our own know-how started from the scratch pad, we believe that our design is not much different from others'. We combine fingerprint verification with AES. Fingerprint method is used in authentication process. That process also generates key for AES encryption/decryption module. We employ AES with 128 bit length in both data and key. The use of fingerprint instead of normal type-in password is expected to be better in security strength since fingerprint is unique and individual biometric. However, the detail of fingerprint verification is omitted in this paper according to its lengthy details as well as its off-track of interest.

The rest of this paper is structured as follows: section II is devoted to the architecture design. Section III is the detail of FPGA implementation together with evaluated results obtained partially. The work is finally concluded in section IV.

II. ARCHITECTURAL DESIGN

The system architecture of the FDE portable hard drive is demonstrated in Fig. 1. The embedded designed system is to be installed in the external hard drive enclosure where the normal ATA drive resides. The system includes a USB interface, AES core, and a fingerprint verification module. The USB interface made use of a custom chip available commercially. Besides the USB interfacing, the main feature of General Programmable Interface (GPIF) is also controlled by the C-51 microcontroller core. The USB interface communicates with a hard drive via ATA command decoder. At that stage we have inserted the data encryption/decryption capability to the data set. The AES core relies on the key sequence provided by the authentication module where the fingerprint is scanned and matched to the stored fingerprint database. The fingerprint scanner and matching is implemented on a custom chip.

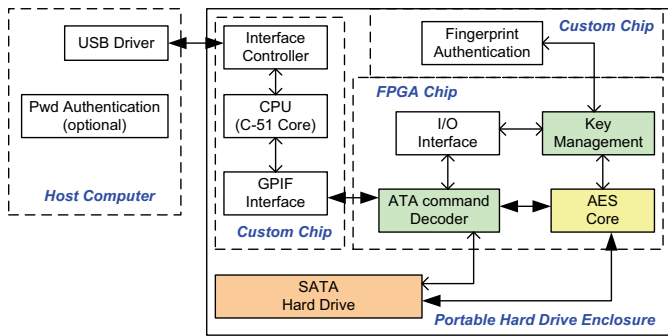


Figure 1. FDE system for a portable hard drive

A. Encryption/Decryption Core

The uses of AES in FDE application do exist in both software and hardware form, however they are not so widely published. Software-based AES is also vulnerable to attacks. In contrast, in the pure hardware implementation is more robust. AES IP cores are also available commercially in the form of both ASIC and FPGAs [2], [3]. To obtain the higher data rate AES (Gbits/second), a technique of parallelization and pipelining [4] can be combined. The implementations are physically secure since tempering by an outside attacker is naturally difficult. It's also a cost-effective solution for many application specific systems. Although our design is individual, it has drawn many useful ideas reported in [5].

The AES encryption/decryption module work on 128 bit data block and in similar manner but in reverse direction. The module comprises similar operations which are: (1) BytesSub transformation or permutation. The operation is a non-linear byte substitution where multiplicative inverse and affine transformation are involved. (2) ShiftRow transformation is a linear diffusion process operates on individual row of a state. (3) MixColumn transformation is also a linear diffusion process. Each data column is multiplied by fixed matrix. And (4) AddRoundKey is the operation operates in a ground field of GF(2). As the engine must iterate for 10 rounds, the roundkey changes its value from round to round. As such, the key scheduling must be designed to cover this requirement. The addition in GF(2) is simple as XOR operation.

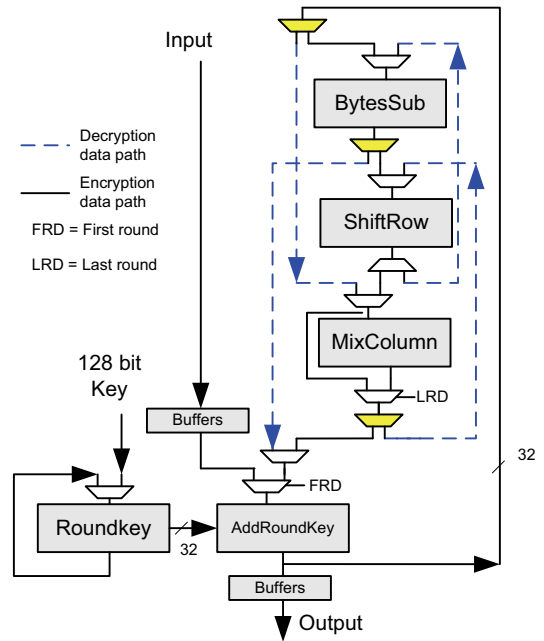


Figure 2. AES Core (32 bit data path)

B. USB Interface

USB interface controller bridges the FPGA and the host computer. Cypress EZ-USB FX2 chip has been adopted. GPIF functions for ATA Protocols such as PIO and UDMA are embedded. When the designed system is plugged into the host computer, EZ-USB FX2 enumerates automatically and downloads both firmware and USB descriptor tables. The host computer will identify EZ-USB FX2 as the development board of EZ-USB FX2. Then EZ-USB FX2 enumerates again as EZ-USB FX2 sample device. If the user passes the authentication check, EZ-USB FX2 enumerates again as the hard drive. If not, the hard disk cannot be remunerated and the hard drive is not recognized.

C. Encryption/Decryption Data Flow

The encryption flow line consists of a data receiver module, a 16-bit to 128-bit width conversion module, an AES encryption module, a 128-bit to 16-bit width conversion module, a FIFO module, and a data sender module.

The plain text data delivered by a host computer is encrypted before storing on a hard drive. In details; (1) The plain text data from the host device is received by the data receiver module. At this module CRC checking is performed. The checking polynomial $g(x) = x^{16} + x^{12} + x^5 + 1$ is used for CRC-16. (2) The 16-bit width plain text is stored and converted to the 128-bit width plain text; (3) The 128-bit width plain text is encrypted by the AES encryption module; (4) The 128-bit encrypted text is converted to the 16-bit width encrypted text; (5) The 16-bit width encrypted text is sent to a 16-bit width FIFO module; (6) The data sender module receives the encrypted text from the FIFO module and sends this encrypted text to store on the hard drive.

The decryption flow line is basically operated in reverse direction to the encryption flow line. Both data flow lines and necessary boxes are shown in Fig. 3.

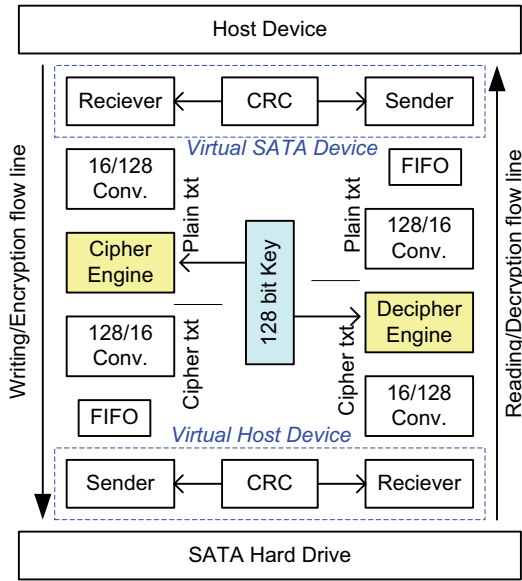


Figure 3. Down streamed and up streamed data flow line

III. FPGA IMPLEMENTATION AND PRELIM RESULTS

All the results are based on simulations from the ISE Test Bench. Individual transformation of both encryption and decryption are simulated using FPGA Spartan 3 families and Xilinx XC3S400 device. By far, only portions of the whole work had been done. We have done mostly the AES core. There are 3 main routines designed to realize the AES algorithm: encryption, decryption and key expansion routines. The complete VHDL code is organized with several modules. Basically, those modules implement the corresponding AES functions as described in previous section. Similar to others', our design criteria are to obtain fastest, smallest and lowest power consumption chip.

TABLE I. AES CORE FPGA UTILIZATION

Resources	AES Core		
	Available	Used	Utilization
Number of Slices	3584 ^a	1040	29%
Number of Slice F/F	7618	914	12%
Number of 4 input LUTs	7618	1910	25%
Number of bonded IOBs	178	302	170%

The design is synthesized and simulated for the above said Xilinx FPGA. The resource utilization is summarized in Table I above. The simulation to verify the function of encryption and decryption is made and the obtained result is illustrated in Fig. 4. Of the obtained data rate of 976 Mbit/sec, we believe that this is enough for USB 2.0 compliant even when the whole designs are integrated. The data rate can be made vary

according to the design architecture. One can lower the data rate by either reducing the gate count or removing some internal pipeline stages.

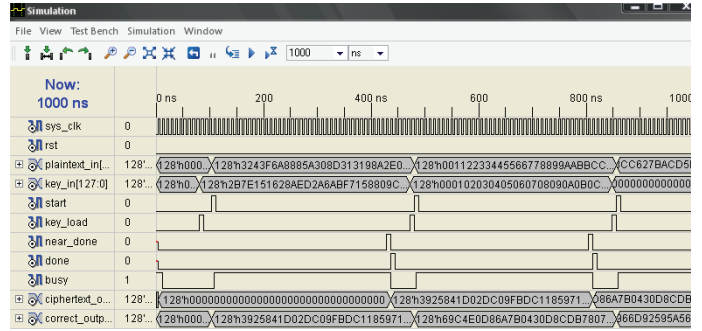


Figure 4. AES core timing simulation and function verification

IV. CONCLUSION

In this paper, we have addressed the implementation of Full Disk Encryption (FDE) system designed for ATA external hard drive. The FDE is superior in term of data security compared to other methods. The system is enclosed in a tiny box together with a hard drive, connect to the host computer via the USB port. Fingerprint scanner as well as matching software is also integrated to the system for biometric authentication. Both commercial chips and our own FPGA are combined to serve full FDE function. Although the system is not yet fully implemented when this manuscript is prepared, the most completed AES core has convinced well the whole system functionality. The obtained throughout of 976 Mbit/sec is by far twice faster than that of USB 2.0 specification. The data rate can be fine-tuned by re-designing the hardware. As we are looking forward to the USB 3.0 complaint architecture, the current design has shown its fairly good potential. In comparison with the system proposed in [6] where AES was also evoked, our data rate is much faster. We used fingerprint identification rather than a password and MEM code lock. However, our architecture and system verifications are partially done and needed further development.

REFERENCES

- [1] NIST Federal Information Processing Standards (FIPS PUB 197) Advanced Encryption Standard (2001, Nov.). [Online]. Available: <http://www.nist.gov/aes>
- [2] CAST, Advanced Encryption Standard Core, available at; <http://www.cast-inc.com/cores/aes/index.shtml>.
- [3] IP Cores, Ultra-Compact, Advanced Encryption Standard Core, available at; <http://www.ipcores.com/AES1.pdf>.
- [4] I.M. Verbauwhede, P.R. Schaumont, and, H. Kuo, "Deign and Performance Testing of A 2.29 Gb/s Rijndael Processor," *IEEE J. of Solid State-Circuit*, Vol. 38, No. 3, March 2003, pp. 569 – 572.
- [5] S. Chantarawong, P. Noo-intara, and S. Choomchuay, "An Architecture for S-Box Computation in the AES," *Proc. of Information and Computer Engineering Workshop 2004 (ICEP2004)*, January 2004, pp.157-162.
- [6] Weiping Zhang, Wenyuan Chen, Jian Tang, Peng Xu, Yibin Li and Shengyong Li, "The Development of a Portable Hard Disk Encryption/Decryption System with a MEMS Coded Lock" *Sensor 2009*, Pages 9300-9331.