REAL-TIME MODE HOPPING OF BLOCK CIPHER ALGORITHMS FOR MOBILE STREAMING

Kuo-Tsang Huang[1], Yi-Nan Lin[2] and Jung-Hui Chiu[1]

[1]Department of Electrical Engineering, Chang Gung University, Tao-Yuan, Taiwan
[2]Department of Electronic Engineering, Ming Chi University of Technology, New Taipei City, Taiwan
d9221006@gmail.com, jnlin@mail.mcut.edu.tw, jhchiu@mail.cgu.edu.tw

ABSTRACT

It has been shown that the encrypted information or ciphertext produced by symmetric-key block ciphers with Electronic codebook mode is vulnerable to ciphertext searching, replay, insertion and deletion because it encrypts each block independently. To compensate for this, each block of the encrypted information should be encrypted dependently. The encrypted information should be operated with a special mode. The operation mode should be changed. This paper analysis what an operational mode of block ciphers needs to feedback exactly and proposes a simple real-time changing operation mode technique that extends the existing mode changing opportunity. The new change operation mode technique considers the sign differences between the intra-feedback information and the public-feedback information, and then adaptively determines the corresponding change operation mode factor for each data block. This mode hopping technique for mobile streaming security is highly suitable for recent block computing in future various environments.

KEYWORDS

Mobile, Stream, Block Cipher, Mode of Operation, Feedback, Hopping

1. INTRODUCTION

Habits of personal records in the past were drawing and making perfect life memory in order to cross over the distance of time or space. Personal records of life memory can sample and collect to retain constantly and be treasured in future moment taste. The intersection of social community and personal information sharing development of stress instantaneity and emphasis on real-time acknowledge, e.g. a GOOD respond. Digital camera surrounding analogy affairs complete digital record, via the mobile communication networks across space restrictions immediately to share personal things with specific groups or the general public.

Security has become an important issue on embedded systems due to the popularity. The typical security requirements across a wide range of embedded systems are user identification, secure network access, secure communications, secure storages, content security and availability. Modern applied cryptography in the communications networks demands high processing rate to fully utilize the available network bandwidth.

Block encryption may be vulnerable to ciphertext searching, replay, insertion, and deletion because it encrypts each block independently. Unfortunately, the non-feedback conventional block ciphers have plaintext-ciphertext pair problem with the disadvantage of limit block region scramble. A disadvantage of this method is that identical plaintext blocks are encrypted into identical ciphertext blocks. Non-feedback Conventional block ciphers do not hide data patterns well. A striking example of the degree to which non-feedback electronic codebook (ECB) mode can leave plaintext data patterns in the ciphertext can be seen when the electronic codebook

DOI : 10.5121/ijwmn.2013.5210

mode is used to encrypt a bitmap image which uses large areas of uniform colour. Block cipher modes of encryption beside the electronic codebook mode have been suggested to remedy these drawbacks.

Figure 1. Everywhere Smart Equipments such as iDevice [1]

Modes of operation are the procedure of enabling the repeated use of a block cipher under a single key. Block cipher modes of encryption beside Electronic codebook (ECB) have been suggested to remedy these drawbacks. In Internet, modern protocols support several operation modes and ciphers to provide the varied choice for suitable various operated environments. For example Secure Socket Layer (SSL) [2] and Internet Protocol Security (IPSec) [3] support multi-cipher and multi-mode [4][5][6][7][8].

The standard modes of operation described in the literature [9][10] provide confidentiality. How to choose an appropriate operation mode? The different mode has the different characters. For example, both of CFB and OFB can be design operating without padding with bit-based size keystream output; both of CBC and CFB can self sync to avoid channel noise error propagation; and both of CFB and OFB encryption and decryption applications need an encryption module only to reach both usages. In addition, only the forward cipher function of the block cipher algorithm is used in both encryption and decryption operations, without the need for the inverse cipher function. However, one disadvantage still exists with these techniques when they are applied to practice encryption: any changing mode in encrypting session and between encrypting blocks requires the re-configuration of crypto module and has no mechanism guaranteeing a smooth transition in such encrypting single session.

Our idea of the improvement security of mobile streaming is making mode of operation hopping like frequency copping in communication. In this paper, a new technique, called feedback exactly technique, is proposed. The technique uses possible options to satisfy any next one of all feedbacks so that can real-time change from any one mode to another, even during the same session. The rest of this paper is organized as follows. Section 2 contains acknowledge of block cipher, multi-mode crypto ASIC, and mobile streaming protections. Section 3 describes an application which applies to dynamically exchange mode of operation. Hopping mode of operation improves mobile streaming security in communication. The idea is based on the proposed technique of feedback exactly as what an operational mode of block ciphers needs,

described in the Section 4. Section 5 contains designs of the smooth changing process and seamless changeable system architecture of crypto module. Finally Section 6 is the conclusions.

2. RELATED WORKS

Symmetric-key ciphers use trivially related cryptographic secret keys for both encryption of plaintext and decryption of ciphertext. The secret encryption key is trivially related to the secret decryption key, in that they may be identical or there is a simple transformation to go between the two secret keys. In practice, the secret keys represent a shared secret between two or more parties that can be used to maintain a private information link.

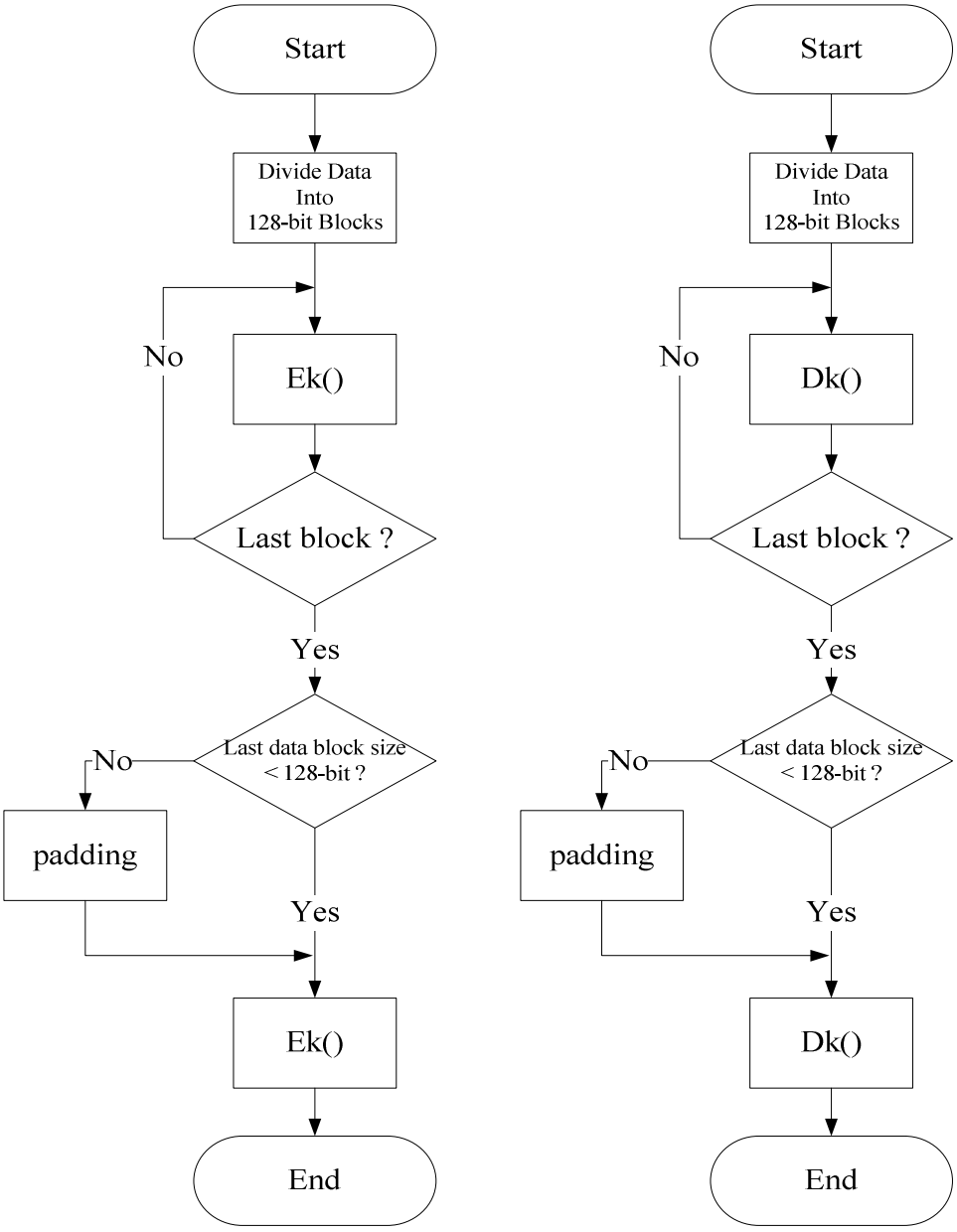


Figure 2. Flow Diagram of Encryption and Decryption for Block Cipher

2.1. Block Ciphers and Padding

The major concept in information security today is to continue to improve encryption algorithms. There are two major types of encryption algorithms for cryptography: symmetric-key algorithms and public-key algorithms. Symmetric-key algorithms also referred to as conventional encryption algorithms or single-key encryption algorithms are a class of algorithms that use the same cryptographic keys for both encryption of plaintext and decryption of ciphertext. It remains by far the most widely used of the two types of encryption algorithms. Symmetric-key encryption algorithms can use either stream ciphers or block ciphers. Block ciphers take a number of bits and encrypt them as a single unit, padding the plaintext so that it is a multiple of the block size. Blocks of 64 bits have been commonly used. The Advanced Encryption Standard (AES) algorithm [14] approved by NIST in December 2001 uses 128-bit blocks. In the normal separation of encryption and decryption, the author separate encryption and decryption processing when receives messages from input port dependent on the target task.

A block cipher works on units of a fixed block size but messages come in a variety of lengths. So that the final block be padded before encryption. Several padding schemes exist. The simplest is to add null bytes to the plaintext to bring its length up to a multiple of the block size, but care must be taken that the original length of the plaintext can be recovered; this is so, for example, if the plaintext is a C style string which contains no null bytes except at the end. Figure 2 is the flow diagram of encryption and decryption for block cipher with padding when last block is not full of bit.

2.2. Mobile Streaming Protections

Business practices of digital content distribution network are most of the structure of Digital Rights Management (DRM) protected [20]. DRM mechanism can protect digital contents against unauthorized user access. From the aspect of the video streaming contents, the encryption mechanism is the protective skill mainly; meanwhile, the encryption methods vary due to the form of video streaming contents. Users receive encrypted digital content through various licensing relationships, which called Rights Object (RO), to distinguish different access control rights. Encryption mechanisms in addition to confidentiality protection in the transmission process, if any change in information will cause decryption failed and, therefore, be able to tell whether a message has been tampered with. Encryption and decryption for protection of information simplify key management issues.

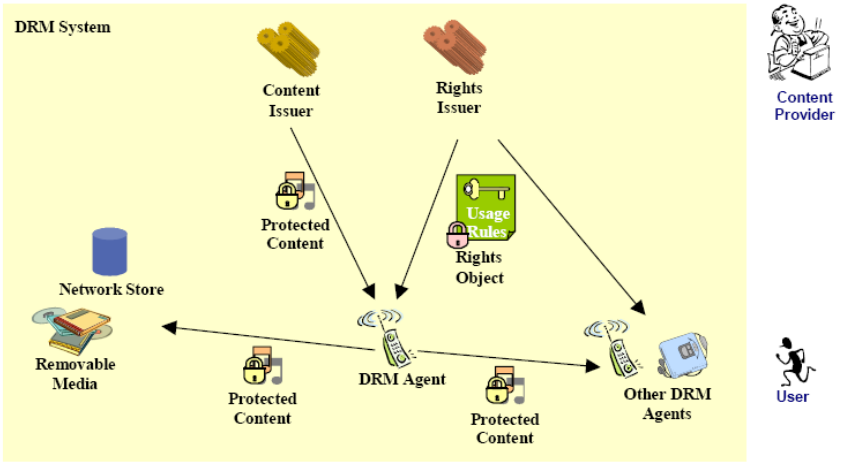


Figure 3. Digital Rights Management Architecture of Open Mobile Alliance [20]

Multimedia security is often used software encryption to ensure the safety of the computing complexity and provide the security requirements of confidentiality. However, the actual use of the classic block cipher (such as AES [14]) to encrypt the entire video stream requires a lot of computing time, and many multimedia applications (such as TV broadcast) require only a low level of security. People need to develop specifically for this Fast encryption target environment.

Transmission of multimedia services with security, such as: secure transmission of visual applications (video clients, video conferencing and other mobile devices ...), variant clients may experience, or low-power devices in the sending or receiving end, such as mobile devices or handheld devices, will encounter problems plagued low-cost devices of receivers [21]. In today's practice, there are two possible solutions: one is a low computational complexity of encryption and decryption algorithms to replace [22], but this will reduce the privacy concerns; the other one is the selective encryption [23][24], encryption is only part of that section of the data, hope to achieve while maintaining safety results. Selective encryption seems to be only part of the contents to ensure the confidentiality, explicitly part of the contents dispersed in the environment. But fortunately the characteristics of video streaming viewing audience must have a smooth flow of quality assurance, an intermittent stream video multimedia content, is contrary to users feeling pain. However, some choose to encrypt which target parts? Alattar etc. [24] proposed a multimedia digital content encryption algorithm for low energy and computing powers handheld devices. For security analysis, they proposed two ciphertext attacks discussed and introduced the technology environment and conditions apply. In the field of multimedia security, encryption for the entire video stream using classical ciphers (such as AES [14]) requires a lot of computing time. The selective encryption algorithms for part of the contents accelerate the existing computing programs. These methods do not provide maximum security but only in the computational complexity of security and balance.

3. THE IMPROVEMENT OF MOBILE STREAMING SECURITY

Cipher modes of operation extend the limit region and scrambling to the post-plaintext message. The formulas of five conventional modes, non-feedback electronic codebook (ECB) mode, cipher block chaining (CBC) mode, output feedback (OFB) mode, cipher feedback (CFB) mode, Counter (CTR) mode, of block cipher for confidentiality could be expressed with the following notations and ciphering Table 1.

Notations:

Ek( )	the Encryption algorithm of the block cipher with session key K
Dk( )	the Decryption algorithm of the block cipher with session key K
Pi	the i-th block of Plaintext with respect to the operation mode
Ci	the i-th block of Ciphertext with respect to the operation mode
Yi	the i-th output State (or keyStream) of the mode like OFB
CTRi	the i-th CounTeR value for the CTR mode of the block cipher
IV	the Initial Value needed for the different operation mode

Table 1. (a) Enciphering for the Conventional Block Cipher Mode of Operation

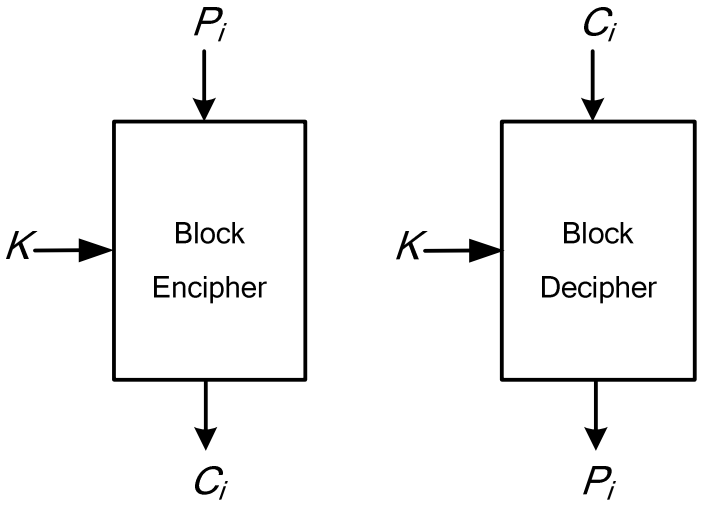
mode of operation	ciphertext formula	extra input
ECB	$C_i = E_k(P_i)$	(not necessary)
CBC	$C_i = E_k(P_i \oplus C_{i-1})$	C_{i-1} with $C_0=IV_1$
CFB	$C_i = P_i \oplus E_k(C_{i-1})$	C_{i-1} with $C_0=IV_1$
OFB	$C_i = P_i \oplus E_k(Y_{i-1})$	Y_{i-1} with $Y_0=IV_2$
CTR	$C_i = P_i \oplus E_k(CTR_{i-1})$	CTR_i with $CTR_0=IV_3$

with $C_0=IV_1$, $Y_0=IV_2$, $CTR_0=IV_3$, and $CTR_i= IV_3+i$

Table 1. (b) Deciphering for the Conventional Block Cipher Mode of Operation

mode of operation	plaintext formula	extra input
ECB	$P_i = D_k(C_i)$	(not necessary)
CBC	$P_i = D_k(C_i) \oplus C_{i-1}$	C_{i-1} with $C_0=IV_1$
CFB	$P_i = C_i \oplus E_k(C_{i-1})$	C_{i-1} with $C_0=IV_1$
OFB	$P_i = C_i \oplus E_k(Y_{i-1})$	Y_{i-1} with $Y_0=IV_2$
CTR	$P_i = C_i \oplus E_k(CTR_{i-1})$	CTR_i with $CTR_0=IV_3$

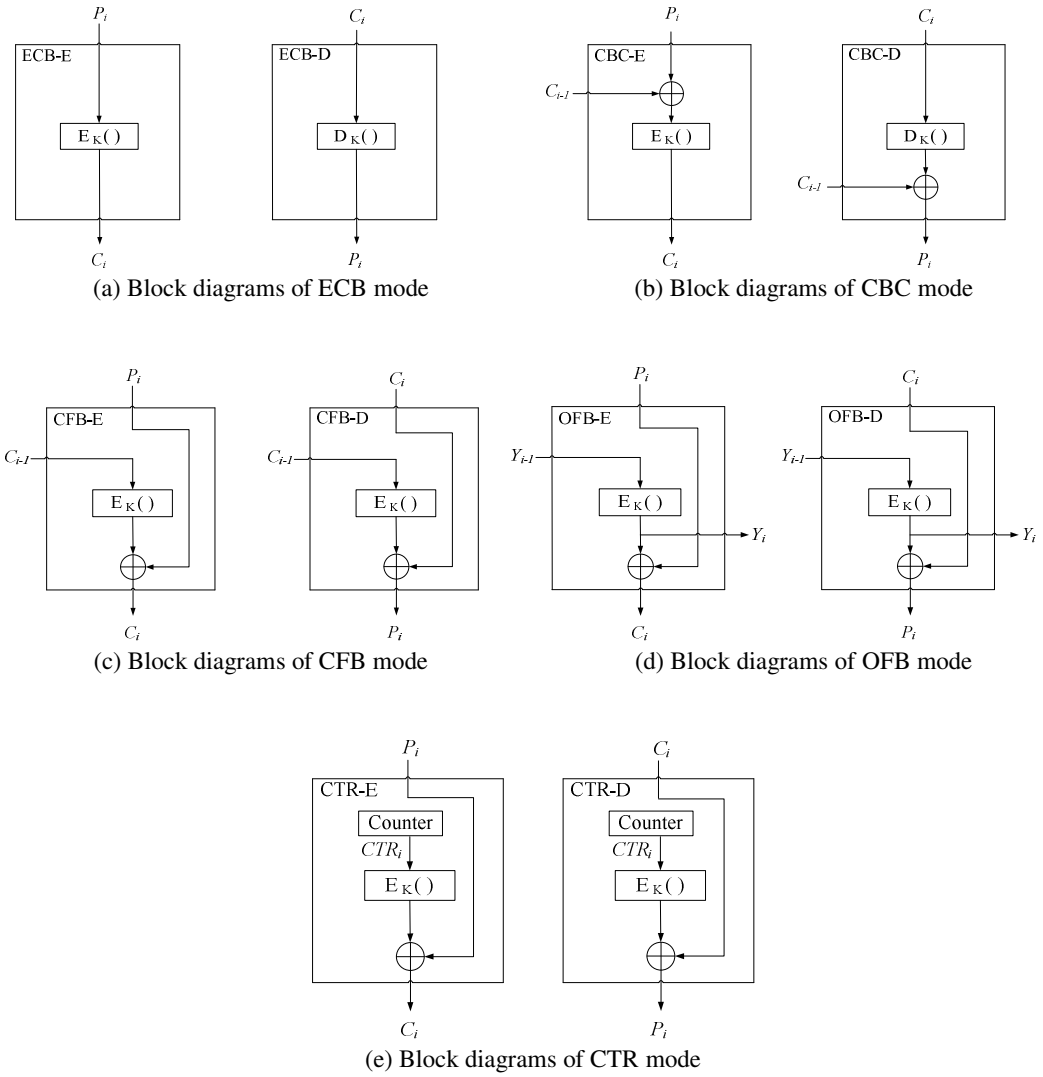
with $C_0=IV_1$, $Y_0=IV_2$, $CTR_0=IV_3$, and CTR i= IV_3+i



(a) Encryption (b) Decryption

Figure 4. Conventional Block Ciphering

The block diagrams of the enciphering and deciphering formulas are shown in Figures 5(a) - (e).



(a) Block diagrams of ECB mode

(b) Block diagrams of CBC mode

(c) Block diagrams of CFB mode

(d) Block diagrams of OFB mode

(e) Block diagrams of CTR mode

Figure 5. Block Diagrams of Different Ciphering Modes

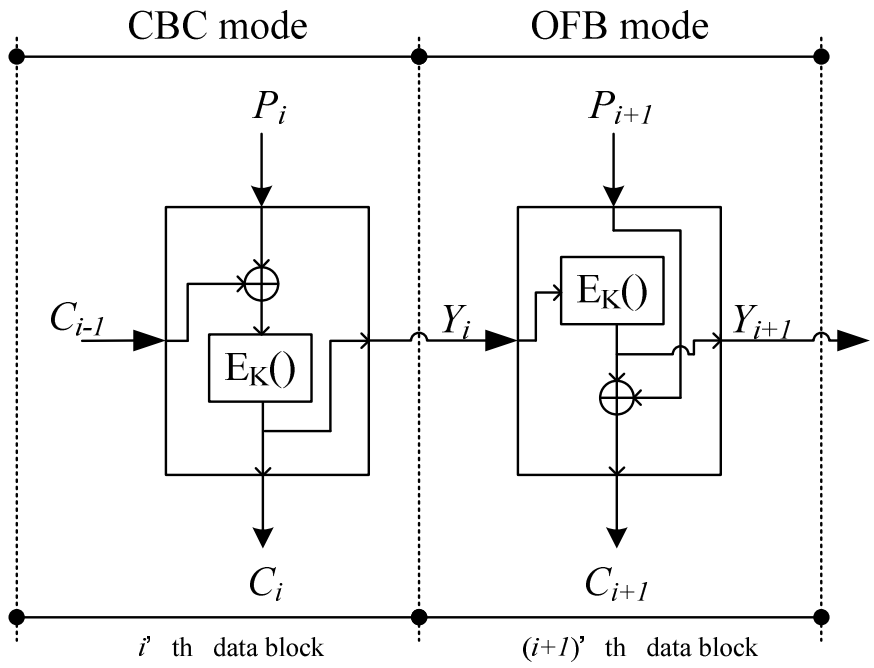
3.1 Problem Descriptions

According the above expressions, except the ECB mode, encipher needs the regular input of plaintext Pi and the extra input (C_{i-1}, Y_{i-1}, or CTR_{i-1}) to perform the ciphering formula to find the output ciphertext C_i. The extra input is not the same for all the modes of operation. This situation will make it not suitable for changing the mode of operation during the ciphering process. In other words, the direct change immediately from one mode to another in a session is impossible. For instance, if the current mode is ECB, it can't change to the OFB mode, because that it does not have the extra input Y_{i-1} which is necessary for the OFB mode to perform correctly.

Based on the conventional ciphertext/plaintext formula, it could be configured to one of the five modes of operation, and performs the ciphering task during the entire session with this fixed mode. After that, it could be reconfigured to another mode of operation for the requirement of the subsequent job. In this section, the author introduce encrypting bursts with different ciphers or different operation modes. One example of the problem is shown in following.

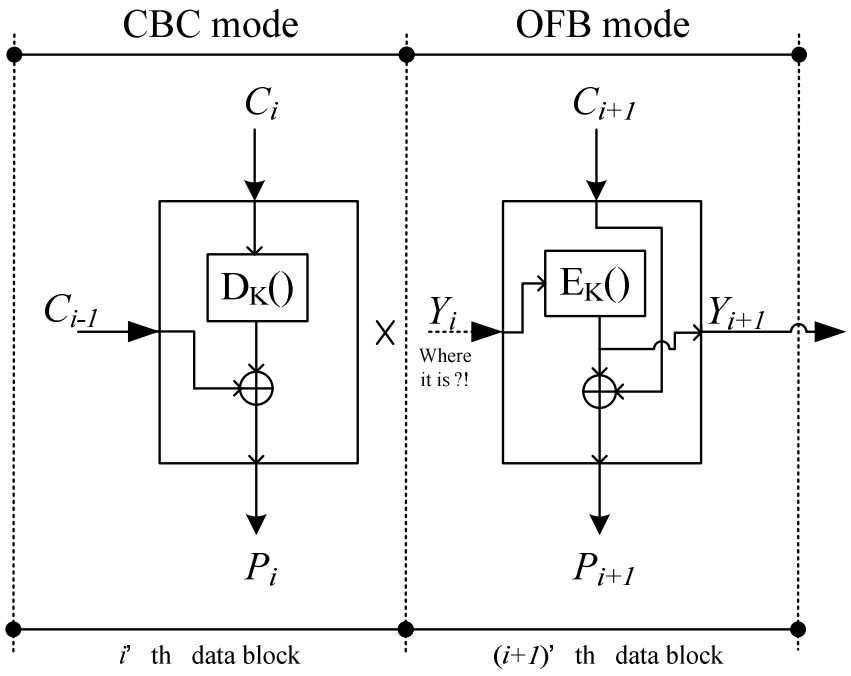
3.2 An Example of Internal Data Feedback Problem

If users want to dynamically change between blocks in a session of encrypt tasks, they must prepare all the feedbacks to suit one of the next block needed. For example, a user must prepare the current ciphertext data block feedbacks, if next block would operate the CBC mode. In current block slot, prepare several possible options for the next block operation needed is a direct solution and can solve the most of feedback problems, but not the special case of internal data feedback problem. Feedback problems include one of the special case the author called internal data feedback problem. The most famous existing example is output feedback problem (i.e., a previous slot block data feedback to the same one single cipher module circularly). The index i is timing slot index in Figure 6. The slot unit of data is block length size.



$C_i=E_K[P_i \oplus C_{i-1}]$ $C_{i+1}=E_K[Y_i] \oplus P_{i+1}$

(a) In encrypting, mode of block operation changing from CBC to OFB



$P_i=D_K[C_i] \oplus C_{i-1}$ $P_{i+1}=E_K[Y_i] \oplus C_{i+1}$, *but where* Y_i *is ?!*

(b) In decrypting, mode of block operation changing from CBC to OFB

Figure 6. An Example of Internal Data Feedback Problem

4. FEEDBACK EXACTLY AS WHAT AN OPERATIONAL MODE OF BLOCK CIPHERS NEEDS

Modern block cipher protocols support five modes of operation, ECB, CBC, OFB, CFB, CTR, to provide the confidentiality for the requirements of different applications. Since each mode has its special plaintext/ciphertext formula with different extra input, it could not change the mode of operation anywhere at any time.

By inspecting formulas and extra inputs in Table 1 and block diagrams in Figure 5, it could be found out that ECB, CBC, and CFB modes have lack of stream variable Si. The assignment of the stream variable Si for ECB, CBC, and CFB modes is necessary. To solve the above inconsistent extra input problem, the union the extra inputs of different modes of operation is set to be the new extra inputs, and be redefined as the previous status $<P_{i-1}, C_{i-1}, Y_{i-1}>$. The status is defined as the 3-tuples vector of the form <plaintext Pi, ciphertext Ci, and internal state Y_i> = $<P_i, C_i, Y_i>$. After finishing the ciphering operation, the previous status $<P_{i-1}, C_{i-1}, Y_{i-1}>$ is updated and replaced the current status $<P_i, C_i, Y_i>$. In this sense, the conventional enciphering formula could be considered as the ciphertext updating formula, and the conventional deciphering formula could be considered as the plaintext updating formula. In order to complete the updating procedure, the update of the **internal feedback exactly state S_i** is introduced here for the different modes of operation. The internal feedback exactly state Si could be found by the general function $g_{mc}( )$ of the mode of operation specified by the mode code m_c.

$$S_i= g_{mc}(k, P_{i-1}, C_{i-1}, Y_{i-1}, CTR_{i-1}) \quad \ldots(1)$$

For the ECB or CBC mode, it does not have the internal feedback exactly state S_i in the conventional form. Hence, the internal feedback exactly state S_i should be defined as to be any variable got from the ECB structure, such as the ciphertext, or that of the next-to-last round. For simplicity, let $S_i = S_{i-1}$.

For the CFB, OFB and CTR modes, they are the stream cipher like. The internal feedback exactly state Si could be defined as to be the keystream variable. Hence, the ciphertext would be expressed as $C_i = P_i \oplus S_i$, and the internal feedback exactly state Si could be expressed as $S_i = E_k (C_{i-1})$, $S_i = E_k (S_{i-1})$, $S_i = E_k (CTR_{i-1})$ for CFB, OFB, CTR, respectively.

Any assignment of S_i for ECB, CBC, and CFB modes could be workable as long as S_i are kept the same at encipher and decipher. For simplicity, the assignments of S_i would be $S_i = S_{i-1}$ for ECB and CBC modes, and $S_i = E_k(C_{i-1})$ for CFB mode, as shown in Table 2.

Enciphering:

$f_{mc}( )$ is the enciphering function of the mode of operation specified by the mode code m_c.

$$C_i = f_{mc}(k, P_i, P_{i-1}, C_{i-1}, S_{i-1}) \quad \ldots(2)$$

Table 2. (a) Enciphering for mode hopping of the improved block cipher mode of operation

operation mode	mode code m_c	enciphering formula (status updating formula)	previous status
ECB	000	$C_i = E_k(P_i)$, $S_i = S_{i-1}$	$P_{i-1}, C_{i-1}, S_{i-1}$
CBC	001	$C_i = E_k(P_i \oplus C_{i-1})$, $S_i = S_{i-1}$	$P_{i-1}, C_{i-1}, S_{i-1}$
CFB	010	$C_i = P_i \oplus S_i$, $S_i = E_k (C_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$
OFB	011	$C_i = P_i \oplus S_i$, $S_i = E_k (S_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$
CTR	100	$C_i = P_i \oplus S_i$, $S_i = E_k (CTR_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$

with $C_0 = IV_1$, $S_0 = IV_2$, $CTR_0 = IV_3$, and $CTR_i = IV_3 + i$

Deciphering :

$f_{mc}^{-1}( )$ is the deciphering function of mode m_c, and is the inverse function of $f_{mc}( )$.

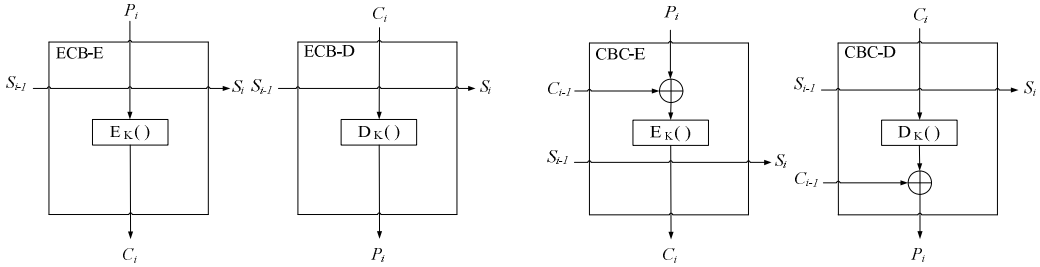
$$P_i = f_{mc}^{-1}(k, C_i, P_{i-1}, C_{i-1}, S_{i-1}) \quad \ldots(3)$$

Table 2. (b) Deciphering for mode hopping of the improved block cipher mode of operation

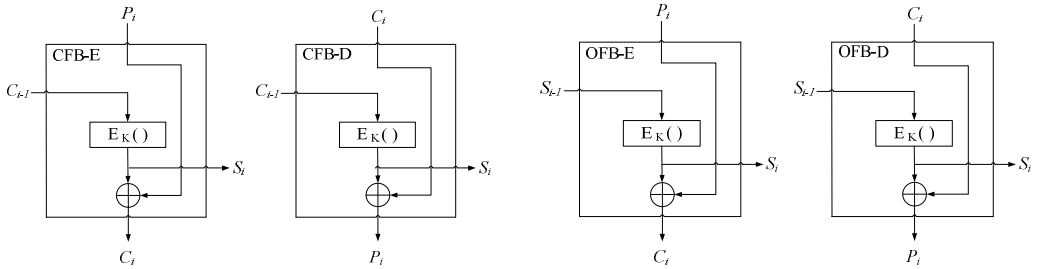
operation mode	mode code m_c	deciphering formula (status updating formula)	previous status
ECB	000	$P_i = D_k(C_i)$, $S_i = S_{i-1}$	$P_{i-1}, C_{i-1}, S_{i-1}$
CBC	001	$P_i = D_k(C_i) \oplus C_{i-1}$, $S_i = S_{i-1}$	$P_{i-1}, C_{i-1}, S_{i-1}$
CFB	010	$P_i = C_i \oplus S_i$, $S_i = E_k (C_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$
OFB	011	$P_i = C_i \oplus S_i$, $S_i = E_k (S_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$
CTR	100	$P_i = C_i \oplus S_i$, $S_i = E_k (CTR_{i-1})$	$P_{i-1}, C_{i-1}, S_{i-1}$

with $C_0 = IV_1$, $S_0 = IV_2$, $CTR_0 = IV_3$, and $CTR_i = IV_3 + i$

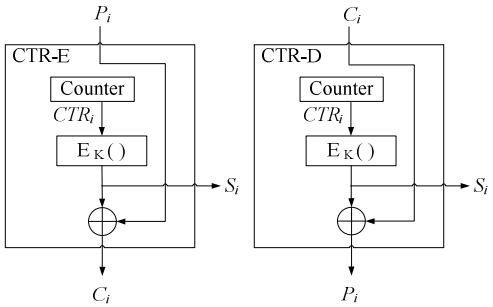
The block diagrams of the improved enciphering and deciphering formulas are shown in Figures. 7(a) - (e).



(a) Improved block diagrams of ECB mode (b) Improved block diagrams of CBC mode



(c) Improved block diagrams of CFB mode. (d) Improved block diagrams of OFB mode.



(e) Improved block diagrams of CTR mode.

Figure 7. Improved Block Diagrams of Different Ciphering Modes

5. APPLY TO REAL-TIME CHANGING OPERATIONAL MODE

Some symmetric ciphers use the same function to perform encryption and decryption applications, some does not. Because of some modes decrypt message using the same algorithm as encrypting, i.e. CFB and OFB. The author defines three levels about module, function, and application to distinguish different usage situations. A crypto cipher module has two functions, an encryption function and a decryption function. E-application is a practical one in message encryption, and D-application is a practical one in message decryption. E-function is a mapping function of transforming information (referred to as plaintext) to make it unreadable to anyone except those possessing special knowledge, usually referred to as a key. D-function is the inverse one of E-function. Table 3 indicates the notations what the author need.

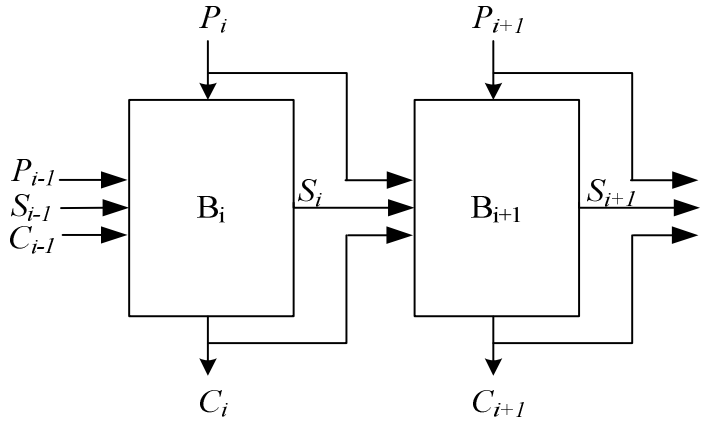
When this simple real-time changing operational mode technique is used for message encryption, the mode change depends on a control sequence. The target sequence is synchronous between encryption and decryption parts and it is pre-shared as a secret. One can generate many different control sequence configuration files at design time for peers which can have a great diversity to change the modules. Before a session of communication begins, the communication parties need to negotiate a control sequence configuration file introduced by adaptive secure protocols such as IPSec. However, the additional cost of a target control sequence can be flexibly reduced according to the practical requirements of security.

If user wants guarantee real-time changing mode of crypto operations, a technique must allow keep one step ahead. The author use AES (Daemen et al, 2002) to illustrate exchange procedures. A secret key $\boldsymbol{K}$ with a 128 bit length is required for the encryption and decryption (denoted by E_K and D_K respectively) of AES (Daemen et al, 2002). Suppose the author want to encrypt a long message sequence, $\boldsymbol{P^T}$. First, the author divide the sequence into several blocks, $\boldsymbol{P_i}$, ***i=1,2...***, each $\boldsymbol{P_i}$ with 128 bit length. Then, the encryption operations are performed by Figure 8(a) and the decryption operations are performed by Figure 8(b). Where $\boldsymbol{S_i}$, for ***i=1,2...,*** is a 128 bit sequence.

Both enciphering and deciphering algorithms are needed implemented within the deciphering improved structure, but only one of them is used for each mode of operation. To save the consuming power, in the deciphering improved structure, the enciphering box is disabled for ECB, CBC modes and enabled for CFB, OFB, CTR modes; and similarly, the deciphering box is enabled for ECB, CBC modes and disabled for CFB, OFB, CTR modes. The author can get that it is flexible, workable, and applicable in modern communication applications.

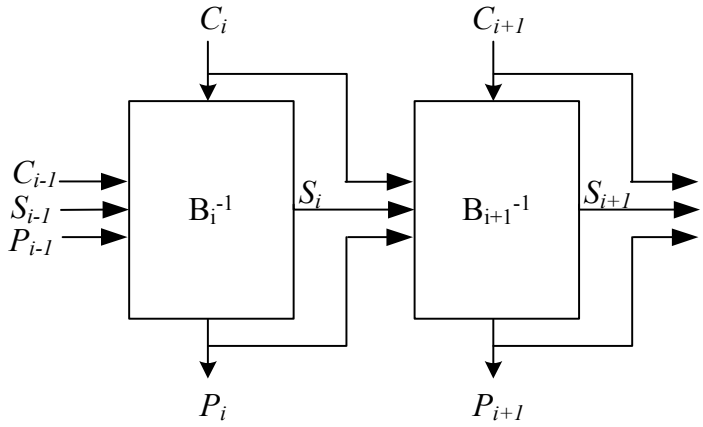
Table 3. Notations

B	A crypto **module** of any one symmetric cipher which can perform both the encryption and decryption **applications**. This module has two **functions**, an encryption function and a decryption function.
$\mathbf{B_i}$	Use B in an encryption application (E-application) at a ***i***'th block slot, i=1,2,…, to encrypt a long message sequence.
$\mathbf{B_i^{-1}}$	Use B in a decryption application (D-application) at a ***i***'th block slot, i=1,2,…, to decrypt a long message sequence.
$\boldsymbol{P^T}$	A message sequence of all the plaintext block data.
$\boldsymbol{C^T}$	A message sequence of all the ciphertext block data.
$\boldsymbol{P_i}$	An ***i***'th block data in a session of task. $\boldsymbol{P_i=D\text{-}application[C_i]=B^{-1}[C_i]}$
$\boldsymbol{C_i}$	An ***i***'th block data in a session of task. $\boldsymbol{C_i=E\text{-}application[P_i]=B[P_i]}$
$\boldsymbol{S_i}$	An ***i***'th output State (or keyStream) of the mode.



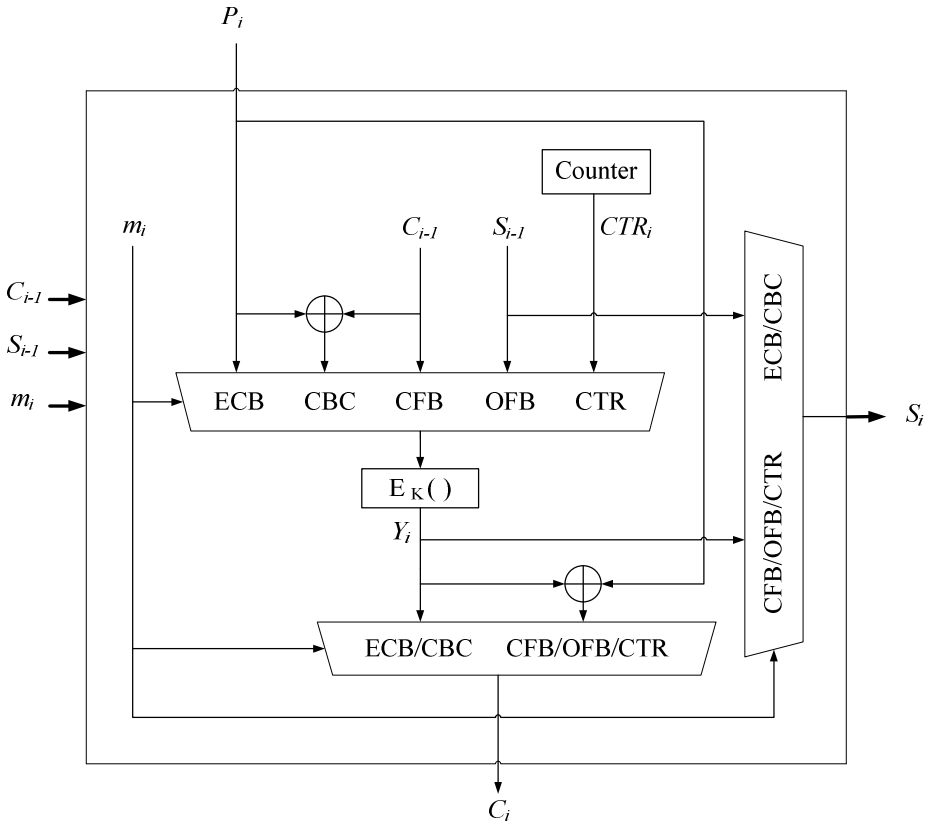
$C_i=B_i[P_i,P_{i-1},C_{i-1},Y_{i-1}]$ $C_{i+1}=B_i[P_{i+1},P_i,C_i,Y_i]$

(a) The encryption operations

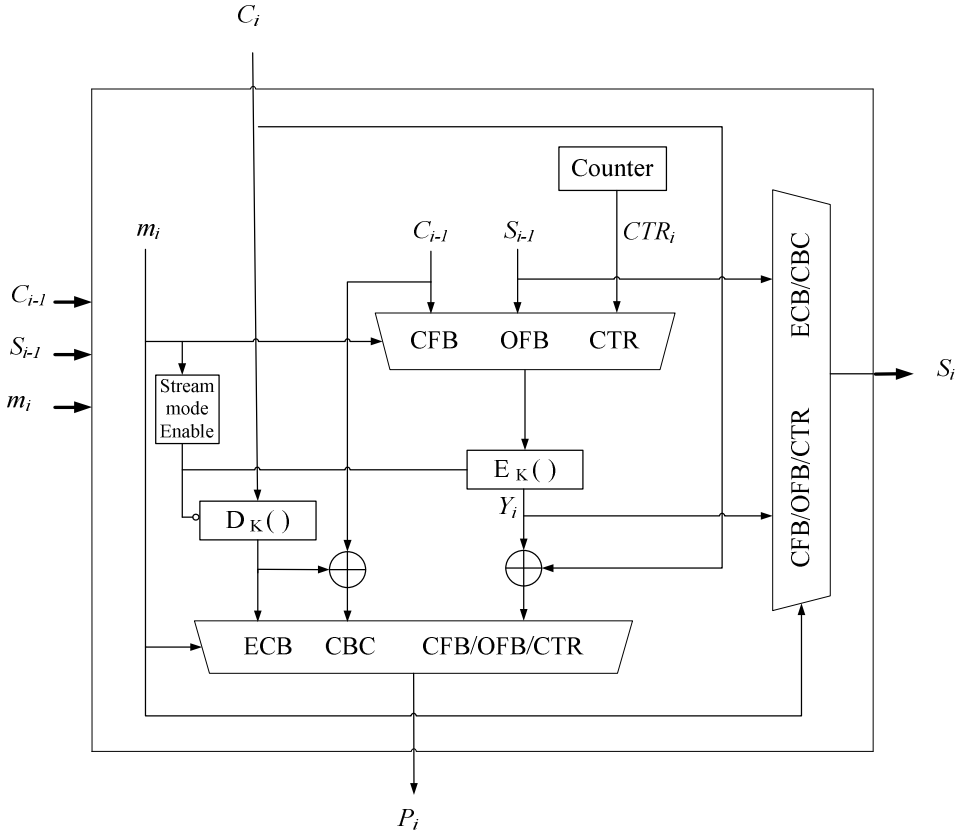


(b) The decryption operations

Figure 8. Real-time Changing Operational Mode



(a) Crypto module $\mathbf{B_i}$



(b) Crypto module $\mathbf{B_i}^{-1}$

Figure 9. System Architecture of Proposed Crypto Module

6. CONCLUSIONS

The design of a real-time hopping mode of operation is proposed in this paper. In this manuscript, the author proposes a seamless changing technique for block ciphers. The changing technique facilitates the smooth implementation of proposed system architecture of dynamic changeable and low-resource hardware, which is appropriate for mobile devices such as a smartphone. According to most of the content in mobile communication are products from social applications or multimedia applications. In the future, people share their own video programs frequently. The author provide a common solution for multi-mode, which is highly suitable for recent block cipher computing using several resources in future various environments, especially a large amount data such as video streaming.

REFERENCES

[1] iDevice. http://programming4.us/multimedia/4782.aspx

[2] SSL 3.0 Specification. http://wp.netscape.com/eng/ssl3/

[3] IPSec Working Group. http://www.ietf.org/html.charters/ipseccharter.html

[4] Block cipher modes of operation. http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation

[5] S. Guilley, P. Hoogvorst, and R. Pacalet, (2007) "A Fast Pipelined Multi-Mode DES Architecture Operating in IP Representation," *The VLSI Journal*, Vol. 40, No. 4, pp.479-489.

[6] L. Luo, Z.G. Qin, and J. Wang, (2009) "The Intelligent Conversion for Different Layers' Block Ciphers," *ICIC Express Letters*, Vol. 3, No. 1, pp.73–77.

[7] C.P. Young, C.C. Chia, Y.B. Lin, and L.B. Chen, (2011) "Fast multi-cipher transformation and its implementation for modern secure protocols," *IJICIC*, Vol. 7, No. 8, pp.4941–4954.

[8] K.T. Huang, J.H. Chiu, and S.S. Shen, (2013) "A Novel Structure with Dynamic Operation Mode for Symmetric-Key Block Ciphers," *International Journal of Network Security & Its Applications*, Vol. 5, No. 1, pp.17-36.

[9] National Institute of Standards and Technology (NIST), NIST. gov - Computer Security Division - Computer Security Resource Center, "Recommendation of block cipher security methods and Techniques," NIST SP800-38.

[10] International Organization for Standardization (ISO), "Information Technology-Security Techniques-Modes of Operation for. an n-bit Block Cipher," ISO/IEC 10116.

[11] William Stallings (2003). *Cryptography and Network Security: Principles and Practices 3rd Edition*. Pearson Education. ISBN 0-13-111502-2.

[12] Alfred J. Menezes, Paul C. van Oorschot and Scott A. Vanstone (1996). *Handbook of Applied Cryptography*. CRC Press. ISBN 0-8493-8523-7.

[13] G.J. Simmons (1999). *Contemporary cryptology: The science of information integrity*. Wiley. ISBN 0-7803-5352-8.

[14] J. Daemen and V. Rijmen (2002). *The Design of Rijndael: AES - the advanced encryption standard*. Springer Verlag.

[15] S. Mangard, M. Aigner and S. Dominikus, (2003) "A highly regular and scalable AES hardware architecture," *IEEE Transaction on Computers*, Vol. 52, No. 4, pp. 483–491.

[16] D. Carlson, D. Brasili, A. Hughes, A. Jain, T. Kisezly, P. Kodandapani, A. Vardharajan, T. Xanthopoulos, and V. Yalala, (2003) "A high performance SSL IPsec protocol security processor," *Proc. ISSCC'03*.

[17] A. Hodjat, P. Schaumont, and I. Verbauwhede, (2004) "Architectural design features of a programmable high throughput AES coprocessor," *Proc. ITCC'04*, pp.498–502.

[18] A. Satoh, S. Morioka, K. Takano, and S. Munetoh, "A compact Rijndael hardware architecture with S-box optimization," *Proc. ASIACRYPT 2001*, Lect Notes Comput Sci 2248, pp.239–254.

[19] P. Saravanan, N. RenukaDevi, G. Swathi, and P. Kalpana, (2011) "A high-throughput ASIC implementation of configurable AES processor," *IJCA*, NSC(3), pp.1–6.

[20] Open Mobile Alliance, Approved Version 2.1, 2008.

[21] K.T. Huang, J. Stu, S.C. Chou, and J.H. Chiu, (2011) "OTP-based hybrid cipher using for mobile e-Learning," *Proc. 2nd World Congress on Computer Science and Information Engineering*, pp. 127–134.

[22] M. Feldhofer and J. Wolkerstorfer, (2007) "Strong crypto for RFID tags - A comparison of low-power hardware implementations," *Proc. IEEE ISCAS'07*, pp. 1839–1842.

[23] M. Grangetto, A. Grosso, and E. Magli, (2004) "Selective encryption of JPEG 2000 images by means of randomized arithmetic coding," *Proc. IEEE International Workshop on Multimedia Signal Processing*, pp. 347–350.

[24] A.M. Alattar and G.I. Al-Regib, (1999) "Evaluation of selective encryption techniques for secure transmission of MPEG video bit-streams," *Proc. IEEE ISCAS'99*, pp340–343.

[25] L. Qiao and K. Nahrstedt, (1998) "Comparsion of MPEG encryption algorithms," *Computers and Graphics*, Vol. 22, No. 4, pp. 1–21.

[26] I. Agi and L. Gong, (1996) " An empirical study of MPEG video transmissions," *Proc. Internet Society Symposium on Network and Distributed System Security*, pp. 137–144.

[27] L. Wu, C. Weaver, and T. Austin, (2001) "CryptoManiac: A Fast Flexible Architecture for Secure Communication," *Proc. IEEE Int. Symp. Comput. Archit.*, pp. 110–119.

[28] S. Laovs, A. Priftis, P. Kitsos, and O. Koufopavlou, (2003) "Reconfigurable crypto process design of encryption algorithms operation modes methods and FPGA integration," *Proc. IEEE Int. Conf. MWSCAS*, pp. 811–814.

Authors

Mr. Kuo-Tsang Huang received B.Sc. from Chung Hua University in 2001 and M.Sc. from Aletheia University in 2003. He is currently studying for the Ph.D. degree in Department of Electrical Engineering of Chang Gung University, Taiwan. He is a member of the International Collaboration for Advancing Security Technology (iCAST). His research interests include wireless network, information security, cryptography, computer architecture issues and technology.

Dr. Yi-Nan Lin received his B.S. degree from the Electrical Engineering Department of National Twiwan Institute of Technology in 1989, M.S. degree in Computer Science & Engineering from the Yuan Ze University in 2000, and Ph.D. degree in Department of Electrical Engineering of Chang Gung University in 2009. He joined the Department of Electrical Engineering at Mingchi University of Technology, Taishan, Taiwan, in 1990. He is now a associate professor in the Department of Electronic Engineering. His current research interests include error-control coding, and digital transmission systems.

Dr. Jung-Hui Chiu received B.S.E.E. from Tatung University in 1971, M.S.E.E. in signal processing and Ph.D. in communication from National Taiwan University in 1973 and 1986 respectively. From 1975 to 1981, he was a research staff with Chunghwa Telecom Labs where he was involved in the research of fiber communications and the microwave systems. During 1981–1986, he was an institutor for the Electronic Department, National Taiwan University of Science and Technology, and was associate professor from 1986 to 2003. He is currently an associate professor in the Department of Electrical Engineering of Chang Gung University, Taiwan. He is a member of IEEE Communications Society, the Chinese Cryptology and Information Security Association (CCISA), and the International Collaboration for Advancing Security Technology (iCAST). His research interests include digital communication systems, wireless communication systems, information security, RFID, hardware security, smart card, and cryptography.