블록 암호 HIGHT를 위한 암・복호화기 코어 설계

손승일*

Design of Encryption/Decryption Core for Block Cipher HIGHT

Seungil, Sonh*

이 논문은 한신대학교 학술 연구비 지원에 의하여 연구되었음

요 약

대칭형 블록 암호 시스템은 암호화와 복호화 과정에서 동일한 암호키를 사용한다. HIGHT 암호 알고리즘은 2010년 ISO/IEC에서 국제표준으로 승인된 모바일용 64비트 블록 암호기술이다. 본 논문에서는 HIGHT 블록 암호 알고리즘을 Verilog-HDL을 이용하여 설계하였다. ECB, CBC, OFB 및 CTR과 같은 블록 암호용 4개의 암호 운영모드를 지원하고 있다. 고정된 크기의 연속적인 메시지 블록을 암・복호화할 때, 매 34클럭 사이클마다 64비트 메시지 블록을 처리할 수 있다. Xilinx사의 vertex 칩에서 144MHz의 동작 주파수를 가지며, 최대 처리율은 271Mbps이다. 설계된 암호 프로세서는 PDA, 스마트 카드, 인터넷 뱅킹 및 위성 방송 등과 같은 분야의 보안 모듈로 응용이 가능할 것으로 사료된다.

ABSTRACT

A symmetric block cryptosystem uses an identical cryptographic key at encryption and decryption processes. HIGHT cipher algorithm is 64-bit block cryptographic technology for mobile device that was authorized as international standard by ISO/IEC on 2010. In this paper, block cipher HIGHT algorithm is designed using Verilog-HDL. Four modes of operation for block cipher such as ECB, CBC, OFB and CTR are supported. When continuous message blocks of fixed size are encrypted or decrypted, the desigend HIGHT core can process a 64-bit message block in every 34-clock cycle. The cryptographic processor designed in this paper operates at 144MHz on vertex chip of Xilinx, Inc. and the maximum throughput is 271Mbps. The designed cryptographic processor is applicable to security module of the areas such as PDA, smart card, internet banking and satellite broadcasting.

키워드

대칭형 블록암호, 암호화, 복호화, HIGHT, 암호시스템

Key word

Symmetric block cipher, Encryption, Decryption, HIGHT, Cryptosystem

* 종신회원 : 한신대학교 (교신저자 : saisonh@hs.ac.kr)

접수일자 : 2011. 12. 07
심사완료일자 : 2012. 02. 15

Open Access http://dx.doi.org/10.6109/jkiice.2012.16.4.778

Ⅰ. 서 론

암호화는 그 의미가 명확하지 않도록 메시지를 부호화하는 과정이며, 복호화는 암호화된 메시지를 본래의 형태로 변환하는 과정이다[1]. 메시지 원래의 형태를 평문(Plaintext)이라하며, 암호화된 형태를 암호문(Ciphertext)이라 부른다.

암호는 과거에는 군사적인 용도 등의 비밀 통신을 위하여 주로 사용하였으나 현재는 인터넷 기반의 사회, 경제 활동의 안전성, 신뢰성, 프라이버시 보호 등을 위한 핵심 기술로서 메일전송, 사용자 인증, 전자상거래 등에 널리 사용되고 있다[2].

대칭형 암호 시스템은 암호화를 위해 사용되는 송신자가 보유하고 있는 키와 복호화를 위해 사용될 수신자의 키가 동일한 시스템이다[3]. 대칭형 블록 암호 알고리즘으로는 DES(Data Encryption Standard), T-DES, IDEA (International Data Encryption Algorithm), SKIPJACK, MISTY 및 AES(Advanced Encryption Standard) 암호 알고리즘 등 다양하게 발표되었다[3,4]. 한편 국내에서도 대칭형 블록 암호 알고리즘인 SEED 및 ARIA 암호 표준안을 발표하였다[2,5,6]. 본 논문에서는 순수 우리 기술로 개발되어 ISO/IEC 국제 표준으로 2010년 최종 승인된 모바일용 64비트 블록 암호화 기술인 HIGHT(HIGh security and light weigHT) 암호 알고리즘의 효율적인 구현에 관해 연구하였다[7].

본 논문의 2장에서는 HIGHT 암호 알고리즘에 대해 소개하고, 3장에서는 HIGHT 암호 알고리즘의 설계에 대해 다루며, 4장에서는 설계된 HIGHT 알고리즘의 검증 및 성능 분석에 대해 다루고, 마지막으로 5장에서 결론을 맺는다.

Ⅱ. 블록 암호 HIGHT 알고리즘

2장은 참고문헌 [7]과 [8]을 기반으로 정리한 것으로 HIGHT 암호는 대칭형 암호로 블록단위로 메시지를 처리하는 블록암호에 속한다. 이는 n비트 단위로 평문을 수신하고 암호키를 적용하여 평문과 동일한 길이의 암호문을 생성한다. HIGHT 암호는 128비트 마스터 키(Master Key)가지고 64비트 평문으로부터 64비트 암호문을 출력한다. 즉, 64비트 블록암호 HIGHT는 128비트 암호키를 이용하여 메시지를 64비트 블록 단위로 암·복호화하는 알고리즘으로 데이터의 기밀성과 같은 기능을 제공하기 위해 사용될 수 있다.

HIGHT의 전체 구조는 일반화된 Feistel 변형 구조로 이루어져 있고 64비트의 평문과 128비트 키로부터 생성된 8개의 8비트 화트닝 키와 128개의 8비트 서브키를 입력으로 사용하여 총 32라운드를 거쳐 64비트 암호문을 생성한다. 그림 1은 HIGHT 암호 알고리즘의 전체 구조를 보여주고 있다.

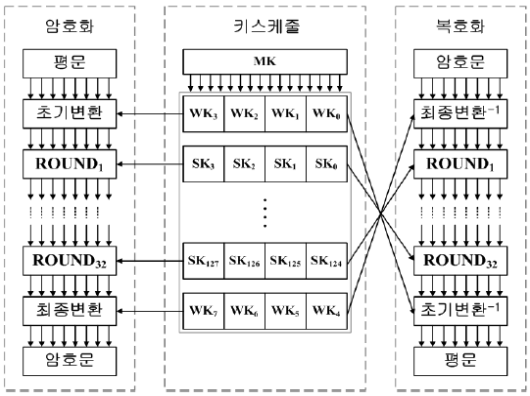

그림 1. HIGHT 암호 알고리즘의 전체 구조
Fig. 1 Overview of HIGHT cipher algorithm

먼저 본 논문에서 사용하는 기본적인 용어를 설명하면 다음과 같다.

1) 64비트 평문과 암호문은 각각 8개의 바이트로 구성되며, 첨자 번호가 큰 것이 MSB에 대응한다.
- 평문 : $P = P_7 \| P_6 \| P_5 ... \| P_0$
- 암호문 : $C = C_7 \| C_6 \| C_5 ... \| C_0$

2) 64비트 라운드함수 입·출력은 8개의 바이트로 구성된다. 표기법은 X_{ij}에서 i는 라운드, j는 바이트 순서를 의미한다.
- $X_i = X_{i,7} \| X_{i,6} \| X_{i,5} ... \| X_{i,0}$ (i=0,1,2...32)

3) 128비트 키(마스터키)는 16개의 바이트로 구성된다.
- $MK = MK_{15} \| MK_{14} \| MK_{13} ... \| MK_0$

4) 라운드함수에 적용되는 라운드키는 서브키 SK_i와 화이트닝키 WK_i가 사용된다.
- SK_i : i =0, 1,...,127(라운드1~라운드32에 적용)
- WK_i : i =0, 1,...,7(초기 및 최종변환에 적용)

5) 산술연산 기호
- ⊞ : 모듈로 2^8 덧셈 연산
- ⊟ : 모듈로 2^8 뺄셈 연산
- ⊕ : XOR 연산
- $A^{\ll S}$: 8비트 값 A에 대한 S 비트 좌측 순환이동

복호화는 암호화 과정의 역 과정이므로 암호화 과정을 위주로 설명하기로 하자.

HIGHT 암호 알고리즘에서 사용하는 화이트닝 키는 마스터 키를 사용하여 아래와 같은 방식으로 생성된다. 즉, 화이트닝 키는 매스터 키의 일부를 사용한다는 것을 의미한다.

$$WK_i = MK_{i+12},\ 0 \le i \le 3 \text{[초기변환에 적용]}$$
$$MK_{i-4},\ 4 \le i \le 7 \text{[최종변환에 적용]}$$

화이트닝 키의 생성 과정을 C 코드 루틴으로 표현하면 아래와 같다.

```
for( i=0; i<4; i++) {
  WK[i] = MK[i+12]; //초기 변환용 화이트닝 키
  WK[i+4] = MK[i]; } //최종 변환용 화이트닝 키
```

그리고 서브키 SKi(SK[i]에 대응됨)를 생성하는 알고리즘을 C 코드 루틴으로 표현하면 다음과 같이 요약된다.

```
for( i=0; i<8; i++) {
  for( j=0; j<8; j++) {
   SK[8+16*i+j   ] = (BYTE) (MK[(j-i)&7]
                          + Delta[16*i+j ]); }
  for( j=0; j<8; j++){
   SK[8+16*i+j+8]= (BYTE) (MK[((j-i)&7)+8]
                          + Delta[16*i+j+8]); }
}
```

위에서 사용되는 Delta[i] 데이터는 128개의 8비트 값들로 구성되어 있으며 LFSR(Left Feedback Shift Register) 연결 다항식 $x^7 + x^3 + 1$로부터 얻어진 값이다.

HIGHT 암호 알고리즘은 라운드 연산 수행시에 사용하는 F 함수는 8비트로 된 256개의 ROM 테이블에 저장되어 있으며 바이트 데이터 X 값에 따라 유효 값을 읽어내도록 구현된다.

이러한 라운드 함수는 라운드 1부터 라운드 32까지 32회 반복 적용된다. 각 i 번째 라운드 함수 Roundi, i=1,... 31는 $X_{i-1} = X_{i-1,7} \| \cdots X_{i-1,0}$을 $X_i = X_{i,7} \| \cdots X_{i,0}$으로 그림 2와 같이 변환한다. 그리고 마지막 라운드 함수인 Round32는 바이트들을 섞지 않고 입력 값 $X_{31} = X_{31,7} \| \cdots X_{31,0}$로부터 출력 값 $X_{32} = X_{32,7} \| \cdots X_{32,0}$를 생성하며, 이후 최종 화이트닝 키를 적용하면 암호화된 결과를 얻게 된다.

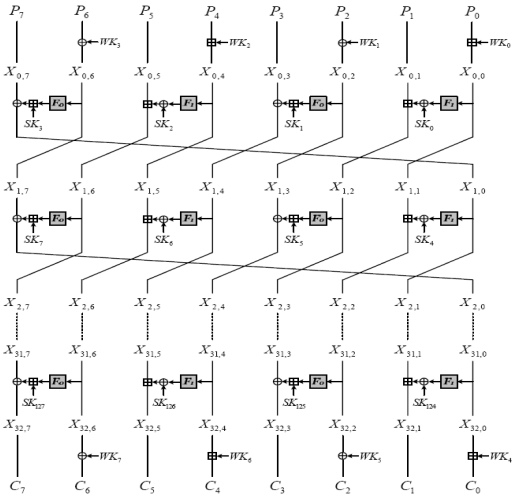

그림 2. HIGHT 암호의 라운드 구조
Fig. 2 Round structure of HIGHT cipher

HIGHT 암호의 복호화 과정은 다음과 같이 요약할 수 있다. 복호화 알고리즘의 초기 변환은 암호화 알고리즘의 최종 변환에 대한 역변환이며, 최종 변환은 암호화 알고리즘에서의 초기 변환에 해당된다. 그리고 라운드 함수 내에서 중간 결과값 X_i와 모듈로 2^8 덧셈이 복호화 알고리즘에서는 중간 결과값 X_i와 모듈로 2^8 뺄셈으로 대체하면 된다.

Ⅲ. HIGHT 암호 알고리즘의 설계

본 논문에서 설계된 HIGHT 블록 암호 알고리즘은 암호 운영모드로 ECB(Electronic CodeBook), CBC(Cipher Block Chaining), OFB(Output FeedBack) 및 CTR(Counter) 모드를 지원하도록 설계하였다[9].

그림 3은 본 논문에서 설계한 HIGHT 블록 암호 알고리즘 코어의 아키텍처를 보여주고 있다. 메시지 버퍼(Message Buffer)는 32비트 4개의 레지스터로 구성되어 있으며, 암호화시에는 평문을 버퍼링하고, 복호화시에

는 암호문을 버퍼링한다. 64비트 단위로 암・복호화를 수행하므로 64비트 데이터가 존재하면 암호화나 복호화의 수행을 시작한다. 그리고 IVreg0과 IVreg1은 초기화 벡터 값을 저장하거나 초기 카운터 값을 저장하기 위해 사용하는 레지스터이다. 초기화 벡터 값인지 카운터 값인지의 여부는 OPMode[1:0] 신호에 의해 결정된다. 또한 HIGHT 블록 암호는 128비트의 키 값을 사용하기 때문에 이를 저장하기 위한 32비트 레지스터 4개가 존재한다. Key0부터 Key3는 매스터 키 값을 저장하는 용도로 사용된다. 메시지 버퍼에 64비트 이상의 데이터가 유효하면 EncStart 신호가 활성화되어 내부의 상태머신인 FSM의 동작이 개시된다. 여기서 Mode 비트가 0이면 암호화 모드로 동작되고, 1이면 복호화 모드로 동작하게 된다. 아울러 암호 운영 모드를 나타내는 OPMode[1:0]의 값과 Mode 비트 값의 조합에 따라 각 라운드에 적합한 적절한 제어 신호를 생성하게 된다. 전처리(Preprocessing) 로직은 라운드가 시작되기 전의 초기 화이트닝 키를 적용하여 연산을 수행함과 동시에 암호 운영 관련 연산을 수행한다. 이후 라운드 코어 블록에서 32회 라운드 연산을 수행한다. 이 과정에서는 2개의 F함수가 사용되는데, 라운드 당 1 클럭 사이클이 수행하기 위해 2개의 F0 ROM과 2개의 F1 ROM을 사용하도록 설계에 반영하였다. 32회의 라운드 연산이 완료되면 후처리(Postprocessing) 과정을 수행한다. 후처리 과정에서는 최종 화이트닝 키를 적용하여 연산을 수행함과 동시에 암호 운영 모드와 관련된 연산을 수행하여 결과 값을 레지스터에 래치한다. 연속되는 암・복호화 데이터가 존재할 경우에는 앞서 설명한 과정을 반복하게 된다. 그리고 후처리가 종료된 64비트 결과 값에 대해 32비트 단위로 2회에 걸쳐 출력한다. 하나의 암・복호화 데이터 블록에 대해 34 클럭 사이클이 소요되며 결과 값을 외부로 출력하는데 2클럭 사이클이 소요된다. 암・복호화가 시작된 이후 최종 결과가 외부로 출력되기까지는 36클럭이 소요되지만, 새로운 암・복호화는 34 클럭 사이클 단위로 수행이 가능하다.

HIGHT 블록 암호 알고리즘은 각 라운드마다 8비트 분할된 4개의 서브 키를 사용한다. 라운드 키는 매스터 키와 Delta 값을 사용하여 4개의 서브 키를 생성하는데, 라운드 0에서는 각 8비트로 분할된 MK[0]부터 MK[3]과 ROM에 읽은 Delta 값을 더하여 서브 키 SK[0]부터 SK[3]을 생성한다. 그림 4에는 해당 라운드에 사용되는 매스터 키 값을 알 수 있도록 표시해주었으며 4라운드 당 8비트씩 좌측 순환하면 각 라운드에 맞는 서브 키를 얻을 수 있다.

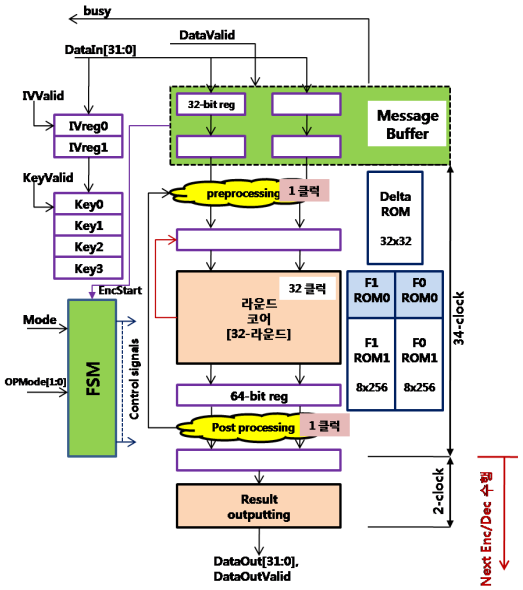

그림 3. HIGHT 블록 암・복호화 코어의 아키텍쳐
Fig. 3 Architecture of HIGHT block encryption/decryption core

그림 4는 암호화시의 서브 키 생성을 위한 과정을 보여주는데, 만약에 복호화를 위한 서브키를 생성하고자 하면 암호화의 역순으로 서브 키를 생성해야 하므로 좌측 순환을 우측 순환으로 변경하고 서브 키 계산시 매스터 키의 읽는 위치만 변경하여 구현할 수 있도록 하였다.

설계된 HIGHT 암호 알고리즘 코어의 입출력 신호는 표1과 같다. 모두 11가지의 I/O 신호를 가진다. Reset 신호는 HIGHT 암・복호화 블록의 코어를 초기화하는 신호이며, clk는 시스템 클럭이다. Mode 신호는 암호화인지 복호화인지를 구별하는 신호이며, OPMode[1:0] 신호는 암호 운영 모드를 정의하는 신호이다. 그리고 입력 데이터를 저장할 때 3개의 valid 신호가 있는 데, 이는 평문이나 암호 입력 데이터의 유효성을 의미하는 DataValid, 암호 키에 대한 데이터의 유효성을 나타내는 KeyValid, 초기화 벡터나 카운터 초기 값의 유효성을 나타내는 IVValid 신호가 있다. 이러한 valid 신호는 32비트 DataIn 입력 신호와 연동하여 사용된다. 즉, DataValid 신호가 1

이면 32비트 DataIn 신호는 Mode 값에 따라 평문이나 암호 데이터가 전달됨을 의미한다.

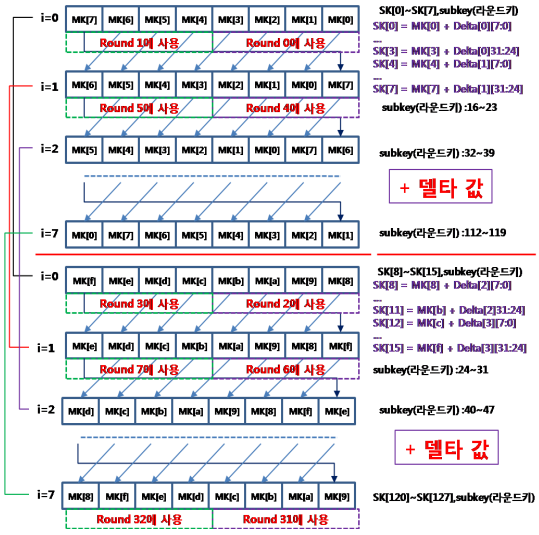

그림 4. 매스터 키와 델타 값을 사용한 서브 키 생성
Fig. 4 Generation of subkey using master key and Delta values

표 1. HIGHT 암호화 알고리즘의 입출력
Table. 1 I/O signals of HIGHT cipher algorithm

신호명	I/O	설명
reset	I	암호화 코어 리셋신호
clk	I	클럭 신호
Mode	I	0:암호화, 1:복호화
OPMode[1:0]	I	암호 운영모드 : 0:ECB, 1:CBC, 2:OFB, 3:CTR모드
DataValid	I	DataIn상에 평문이나 암호문 데이터가 유효함
KeyValid	I	DataIn상으로 유효한 매스터 키가 존재함
IVValid	I	DataIn상에 초기값이나 카운터값이 유효함
DataIn[31:0]	I	32비트 데이터 입력
Busy	O	암호화 코어의 메시지 버퍼가 full 임을 의미
DataOutValid	O	암호화 코어의 출력 데이터가 유효함
DataOut[31:0]	O	32비트 암호화 코어의 출력 데이터

Ⅳ. HIGHT 알고리즘의 검증 및 성능 분석

HIGHT 암호 알고리즘의 검증을 위해 먼저 참고 문헌 [8]에서 제공된 C 소스 코드를 분석하고 참고 문헌 [7]에서 제시된 테스트 케이스를 활용하였다. Verilog HDL을 사용하여 본 논문에서 설계한 HIGHT 암호 알고리즘의 코어에 참고 문헌 [7]에서 제시된 테스트 케이스의 C 소스 코드 수행 값과 비교를 수행하여 양 쪽의 결과 값이 동일하면 정상적인 설계가 이루어졌다고 판단할 수 있다. 그림 5는 본 논문에서 설계한 HIGHT 암호 알고리즘의 암호화 검증을 수행한 시뮬레이션 파형을 보여준다. Mode신호가 0이고, 사용한 암호화 키 값, 초기 벡터 값 및 사용된 평문 입력이 그림에 나타나 있다. 암호 운영모드는 CBC 모드를 적용한 것으로 암화화 수행을 완료한 최종 결과는 “b5fbc1ce” 및 “c85595d9”임을 알 수 있다. 그리고 이러한 결과 값은 앞에 설명한 바와 같이 C 프로그램의 수행 결과와 동일함을 알 수 있다.

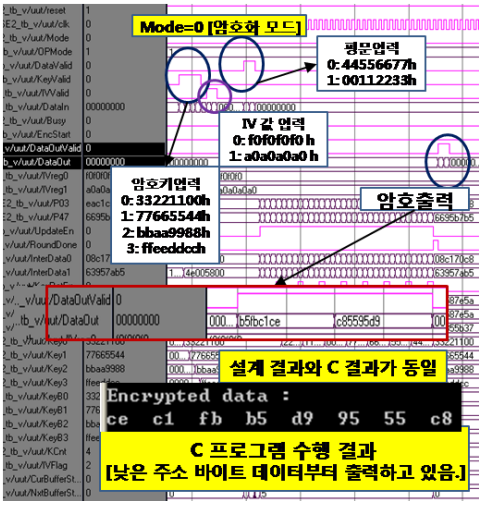

그림 5. HIGHT 암호 알고리즘의 암호화 시뮬레이션
Fig. 5 Simulation of encryption of HIGHT block cipher

그림 6은 Mode 신호가 1인 HIGHT 암호 알고리즘의 복호화 과정에 대한 시뮬레이션을 보여주고 있다. 이 과정은 암호화 과정의 역 과정으로 암호화된 입력 값으로부터 평문을 도출해내는 과정이다. 암호 키, IV 벡터 값은

동일하고 입력은 그림 5의 암호화된 결과 값을 사용하였다. 복호화된 최종 결과는 그림 5에서 사용한 최초의 평문 입력과 동일한 값이 되는 것을 확인할 수 있었다.

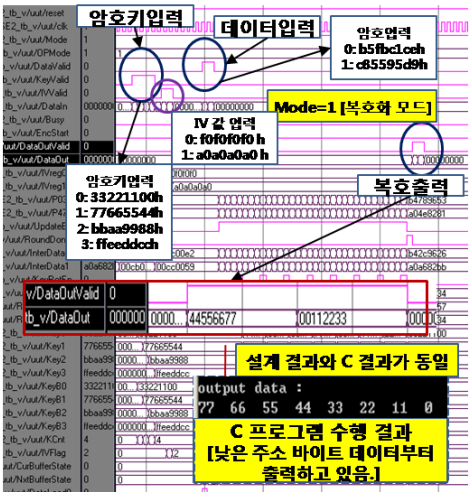

그림 6. HIGHT 암호 알고리즘의 복호화 시뮬레이션
Fig. 6 Simulation of decryption of HIGHT block cipher

그림 7은 HIGHT 암호 알고리즘의 암·복호화 수행에 대한 타이밍도를 보여주고 있다. 결과 값은 매 34클럭 사이클 이후에 출력되고, 34클럭 사이클 간격으로 새로운 메시지 블록에 대해 암·복호화를 수행할 수 있도록 설계하였다.

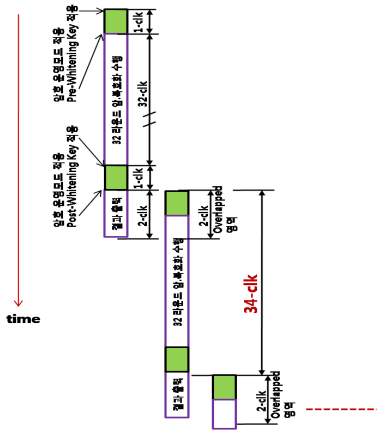

그림 7. HIGHT 암호 알고리즘의 파이프라인 수행
Fig. 7 Pipeline execution of HIGHT block cipher

따라서 매 34 클럭 사이클 간격으로 암·복호화된 결과 값을 출력할 수 있음을 알 수 있다.

표 2는 본 논문에서 설계된 HIGHT 암호 알고리즘의 설계 결과를 정리한 것이다. 현재까지 HIGHT 암호 알고리즘과 관련하여 발표된 설계 논문이 없는 관계로 본 논문의 설계 결과를 정리하였다. 본 논문의 HIGHT 암호 알고리즘은 4개의 암호 운영모드를 지원하며 입출력 데이터 버스는 32비트를 사용하며 34클럭 사이클 단위로 새로운 암·복호 데이터를 처리할 수 있으며 최종적인 동작 주파수는 Xilinx사의 Vertex 칩에서 144MHz인 것을 확인하였다. 따라서 최대 암·복호율은 271Mbps가 된다.

표 2. HIGHT 암호 알고리즘의 설계 결과
Table. 2 Result of Design for HIGHT block cipher

항 목	설 명
지원 암호 운영모드	ECB, CBC, OFB, CTR
블록 처리 단위	64비트
동작 주파수	144MHz
입출력 데이터 버스	32 비트
사용된 슬라이스 수	1,122개(FPGA)
슬라이스 플립플롭 수	843개(FPGA)
단위 암호 블록당 클럭 사이클 수	36 클럭 사이클
연속 암호블록 입력 간격	34 클럭 사이클
최대 암·복호율	271Mbps @ECB, CBC, OFB, CTR

한편 논문 [10]은 0.35um CMOS 표준 셀 라이브러리 이용하여 구현하였는데, 80MHz에서 동작되며 약 150 Mbps의 성능을 갖는 것으로 알려져 있다. 따라서 본 논문에서 제안한 HIGHT 암호 하드웨어 구현이 121 Mbps 더 빠른 속도로 동작한다.

V. 결 론

본 논문에서는 128비트 마스터 키를 적용하여 64비트 단위로 암·복호화를 수행하는 HIGHT 블록 암호 코어를 설계하였다. 설계된 HIGHT 암호 알고리즘은 144MHz에서 동작하며, 초당 최대 271Mbps의 처리 속도를 가진다. 이러한 HIGHT 블록 암호 알고리즘은 PDA,

스마트 카드, 인터넷 뱅킹, 위성 방송 등의 보안 장치로서 활용이 가능하고, 특히 RFID 등과 같이 초고속, 초경량, 저전력을 요구하는 유비쿼터스 분야에서 활용이 기대된다[7].

향후 본 연구 결과를 활용하면 국내에서 표준화된 블록 암호 표준안인 SEED 및 ARIA 알고리즘을 통합한 통합 블록 암호 모듈의 설계에도 도움이 될 것이라 사료된다.

참고문헌

[1] Charles P. Pfleeger, *Security in Computing,* 2nd Edition, Prentice-Hall International, Inc. 2000

[2] 하성주, 이종호, "블록 암호 ARIA를 위한 고속 암호기/복호기 설계", 전기학회논문지, 제 57권 9호, pp.1652-1659, Sep. 2008

[3] 손승일, 최병윤, 강민구, "암호칩 기술 동향", 인터넷정보학회지, Vol.1 No.2, pp89-93, Dec. 2000

[4] 최병윤, "AES Rijndael 알고리즘용 암호 프로세서의 설계, 한국통신학회논문지, Vol.26 No.10B, pp.1491-1500, Oct. 2001

[5] 한국정보호센터, *128 비트 블록 암호 알고리즘 (SEED) 개발 및 분석 보고서,* Dec. 1998

[6] 최병윤, 김진일, "SEED 암호 보조 프로세서의 CPLD 구현", 한국신호처리 및 시스템 학회논문지, 제1권 1-2호, pp.177-185, Oct. 2000

[7] 한국정보통신기술협회, *64비트 블록 암호 HIGHT (TTAK.KO-12.0040/R1),* Dec. 2008

[8] 한국정보통신기술협회, *HIGHT_KISA.c,* C 소스코드, 2008

[9] 한국정보통신기술협회, *블록암호 알고리즘 SEED의 운영모드(TTAS.KO-12.0025),* Dec. 2003

[10] 박해원, 신경욱, "64비트 블록암호 알고리듬 HIGHT의 효율적인 하드웨어 구현", 한국해양정보통신학회 논문지, 제15권 9호, pp.1993-1999, Sept. 2011

저자소개

손승일(Seung-Il Sonh)

1989년 연세대학교 전자공학과(학사)
1991년 연세대학교 대학원 전자공학과(석사)
1998년 연세대학교 대학원 전자공학과(박사)
1998～2002년 호남대학교 컴퓨터공학과 조교수
2008～2009년 미국 미시간공과대학 방문교수
2002년～현재 한신대학교 정보통신학과 교수
※관심분야 : ATM 통신 및 보안, ASIC 설계