The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 21, No. 2, pp.1-6, Apr. 30, 2021. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2021.21.2.1
JIIBC 2021-2-1

CCTV 영상보안 위한 AES 암호 알고리듬의 효율적인 구현

An Efficient Implementation of AES Encryption Algorithm for CCTV Image Security

강민섭*

Min-Sup Kang*

요 약 본 논문에서는 C# 언어를 이용하여 CCTV 영상보안 시스템의 효율적인 구현을 제안한다. 제안한 방법에서는 AES 알고리듬의 각 라운드 과정에서 요구되는 지연시간의 최소화를 위한 합성체 기반의 S-Box를 설계하고, 이를 기반으로 한 영상보안 시스템을 $GF(((2^2)^2)^2)$ 상에서 구현한다. 또한, 메모리 공간의 최소화를 위해서 각 라운드 변환 및 키 스케쥴링 과정에서 필요한 S-Box를 공동으로 사용하도록 설계한다. 성능평가를 통하여 기존의 방법 보다 제안한 방법이 보다 효율적임을 확인하였다. 제안한 CCTV 영상보안 시스템은 Visual Studio 2010을 사용하여 C# 언어로 구현하였다.

Abstract In this paper, an efficient implementation of AES encryption algorithm is presented for CCTV image security using C# language. In this approach, an efficient S-Box is first designed for reducing the computation time which is required in each round process of AES algorithm, and then an CCTV image security system is implemented on the basis of this algorithm on a composite field $GF(((2^2)^2)^2)$. In addition, the shared S-Box structure is designed for realizing the minimized memory space, which is used in each round transformation and key scheduling processes. Through performance evaluation, it was confirmed that the proposed method is more efficient than the existing method. The proposed CCTV system in C# language using Visual studio 2010.

Key Words : CCTV, image security, AES encryption, composite field, S-Box

Ⅰ. 서 론

CCTV(Closed Circuit Television)는 영상정보를 특정의 목적으로 사용하기 위하여 주로 IP 기반의 네트워크로 구성된다. 하지만 현재 CCTV 보안은 너무 허술하여 네트워크 접근에 따른 해킹이 가능하여 사생활 침해, 개인정보 유출이나 침해, 데이터 위변조 등 다양한 위협에 노출되어 있다. 이와 같이 디지털 정보를 보호하기 위하여 암호화 기술을 이용한 CCTV 보안 시스템 설계 방법들이 제안되었다[1].

AES(Advanced Encryption Standard)는 암호화키와 복호화키가 동일한 대칭형 암호시스템이다. 기존의 대칭키 방식의 표준 DES(Data Encryption Standard)가 컴퓨터의 계산능력 향상으로 인한 안전성 저하 문제

*정회원, 안양대학교 컴퓨터공학과
접수일자 2021년 3월 10일, 수정완료 2021년 3월 30일
게재확정일자 2021년 4월 9일

Received: 10 March, 2021 / Revised: 30 March, 2021 / Accepted: 9 April, 2021
*Corresponding Author: mskang@anyang.ac.kr
Dept. of Compter Engineering, Anyang University, Korea

가 대두되어 미국 상무부 기술표준국(NIST)에서 차세대 암호표준 과정을 통하여 벨기에의 RIJNDAEL을 최종 AES로 선정하였고, AES는 효율, 보안, 성능, 구현, 유연성과 같은 다양한 면을 고려할 때 기존 암호화 알고리즘에 비해 성능이 뛰어나다[2].

이러한 암호 시스템에 수반되는 실행 시간과 자원사용량을 최소화하여 성능을 향상시키기 위한 방법이 연구되고 있다[2-3].

AES 암호 알고리듬에 있어서 SubBytes (InvSubBytes) 연산은 많은 양의 메모리가 필요할 뿐만 아니라, 수행시간도 많이 소모되는 모듈이다. S-Box를 구현하는 많은 방법이 제안되었으며, 가장 기본적인 연산방법은 S-Box를 LUT(Look-Up Table) 방식으로 구현하는 것이다. 이 방식은 암호화 연산을 위한 S-Box와 복호화 연산을 위한 InvS-Box가 독립적으로 필요하며, 고정된 설계 구조로 인하여 많은 지연(delay)이 발생하게 된다[4].

이와 같이 문제점들을 개선하기 위한 효율적인 S-Box를 기반으로 한 효율적인 AES 암호 알고리듬의설계가 요구된다[5].

본 논문에서는 C#을 이용하여 효율적인 S-Box기반의 CCTV 영상보안 시스템 구현을 제안한다. 제안한 방법에서는 AES 알고리즘의 라운드 과정에서 가장 큰 시간을 소모하는 S-Box의 개선된 구조를 설계하고 제안한다.

II. CCTV 영상 보안시스템

1. 시스템의 구성도

그림 1은 영상 보안시스템의 전체 구성도를 나타낸다. 그림 1에서 카메라 모듈에서 촬영한 영상을 현재 프레임과 이전 프레임을 비교하여 두 프레임간의 차이를 비교하여 두 프레임간의 픽셀 차를 구하여 저장한다. 두 프레임간의 픽셀 차이를 전체 해상도에서 적당한 크기의 블록으로 잘라내어 블록의 위치를 왼쪽에서 오른쪽으로 정렬시킨다. 이 데이터를 AES 암호화 알고리즘을 이용하여 암호화 하여 DB에 저장한다[6].

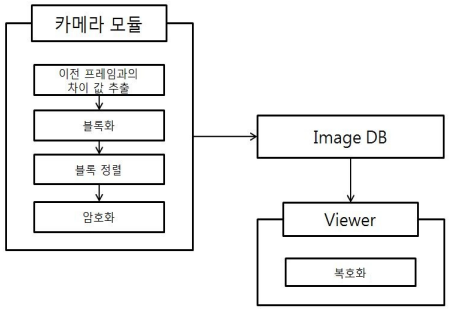

그림 1. 영상 보안 시스템 구성도
Fig. 1. Configuration of image security system

저장된 데이터를 뷰어로 전송 후 뷰어에서 해당 데이터를 복호화 및 역처리를 통해 차이 값을 데이터로 복구한다.

2. AES 암호 알고리듬

AES 알고리즘은 대칭형 암호화 알고리즘이며 4×4 바이트 행렬인 128비트의 데이터 블록에 대한 SubBytes, ShiftRow, MixColumns 및 AddRoundKey 등의 연산을 반복 수행하여 암호문을 생성한다. 그림 2는 AES 암호 알고리듬의 수행과정을 나타낸다[2].

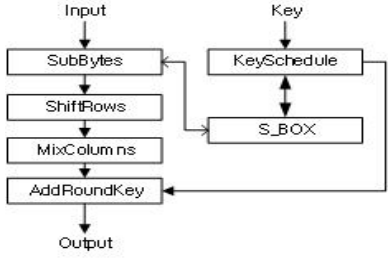

그림 2. AES 암호 알고리즘
Fig. 2. AES encryption algorithm

복호화는 암호화의 역순으로 진행되고 마찬가지로 최종 라운드에서 Inverse MixColumns 연산을 수행하지 않는다.

그림 3은 LUT 기반의 S-Box를 구현하기 위한 C# code를 나타낸다.

AES 알고리듬에서 S-Box는 8비트의 입력 값에 대해 8비트의 출력 값을 가지며, 비선형 연산을 실행한다. 128비트 알고리즘의 경우 SubBytes 연산에 16개, KeySchedule 에는 4개의 S-Box가 필요하므로 S-Box는 암호화에서 매우 큰 용량의 메모리가 요구된다.

본 논문에서는 메모리 공간의 최소화를 위해서 각 라운드 변환 과정 및 키 스케쥴링 과정에서 요구되는 S-Box를 공동으로 사용하는 구조로 설계하고 구현한다 (그림 2 참조).

```
private string s_box(string value) {
    string[,] sbox = newstring[16, 16] {
{"63", "7C", "77", "7B", "F2", "6B", "6F", "C5", "30", "01", "67", "2B", "FE", "D7", "AB", "76"},
{"CA", "82", "C9", "7D", "FA", "59", "47", "F0", "AD", "D4", "A2", "AF", "9C", "A4", "72", "C0"},
{"B7", "FD", "93", "26", "36", "3F", "F7", "CC", "34", "A5", "E5", "F1", "71", "D8", "31", "15"},
{"04", "C7", "23", "C3", "18", "96", "05", "9A", "07", "12", "80", "E2", "EB", "27", "B2", "75"},
{"09", "83", "2C", "1A", "1B", "6E", "5A", "A0", "52", "3B", "D6", "B3", "29", "E3", "2F", "84"},
{"53", "D1", "00", "ED", "20", "FC", "B1", "5B", "6A", "CB", "BE", "39", "4A", "4C", "58", "CF"},
{"D0", "EF", "AA", "FB", "43", "4D", "33", "85", "45", "F9", "02", "7F", "50", "3C", "9F", "A8"},
{"51", "A3", "40", "8F", "92", "9D", "38", "F5", "BC", "B6", "DA", "21", "10", "FF", "F3", "D2"},
{"CD", "0C", "13", "EC", "5F", "97", "44", "17", "C4", "A7", "7E", "3D", "64", "5D", "19", "73"},
{"60", "81", "4F", "DC", "22", "2A", "90", "88", "46", "EE", "B8", "14", "DE", "5E", "0B", "DB"},
{"E0", "32", "3A", "0A", "49", "06", "24", "5C", "C2", "D3", "AC", "62", "91", "95", "E4", "79"},
{"E7", "C8", "37", "6D", "8D", "D5", "4E", "A9", "6C", "56", "F4", "EA", "65", "7A", "AE", "08"},
{"BA", "78", "25", "2E", "1C", "A6", "B4", "C6", "E8", "DD", "74", "1F", "4B", "BD", "8B", "8A"},
{"70", "3E", "B5", "66", "48", "03", "F6", "0E", "61", "35", "57", "B9", "86", "C1", "1D", "9E"},
{"E1", "F8", "98", "11", "69", "D9", "8E", "94", "9B", "1E", "87", "E9", "CE", "55", "28", "DF"},
{"8C", "A1", "89", "0D", "BF", "E6", "42", "68", "41", "99", "2D", "0F", "B0", "54", "BB", "16"};
    r = Convert.ToInt32(value.Substring(0, 1), 16);
    c = Convert.ToInt32(value.Substring(1, 1), 16);
    result = sbox[r, c];      return result;
}
```

그림 3. LUT 기반의 S-Box code
Fig. 3. LUT based S-Box code

III. 효율적인 AES S-Box 설계

AES 알고리듬에서의 S-Box 구현은 Galois Field (GF) 상에서 연산을 수행한다. 1-바이트를 $GF(2^8)$)의 다항식으로 표현하고, 이때 각 원소들은 $GF(2^2)$의 형태로 변환시켜 사용한다. AES-128에서 SubByte 연산은 1바이트를 $GF(2^8)$ 상에서의 역원을 구한후 아핀 변환을 행하고, 8비트씩의 치환 연산을 반복하여 16회 수행한다[2].

SubBytes 과정은 유한체 $GF(2^8)$ 상에서 곱셈 역원 연산을 실행한 후, 아핀 변환(Affine Transformation)을 하게 된다. 그리고 InvSubBytes 과정은 이와 반대로 수행하게 된다. 그림 4는 곱셈역원(multiplicative inverse) 연산이 SubBytes와 InvSubBytes에서 동시에 사용하는 과정을 나타낸다[5].

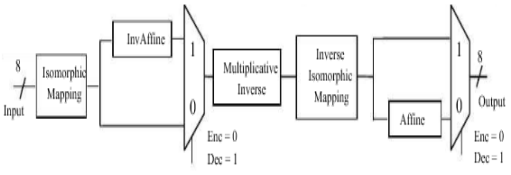

그림 4. 합성체에서 SubBytes와 InvSubBytes변환 과정
Fig. 4. SubBytes and InvSubBytes transformation in composite field

그림 4에서 Enc = 0 일 경우는 Mux의 0 입력이 출력으로 연결되어 암호화 과정이 수행되어 아핀변환 후에 연산 결과가 출력된다. 복호화 경우는 입력에 대해 아핀 변환을 먼저 수행한 후에, 곱셈역원 연산 과정을 거치게 된다. LUT 기반의 AES S-Box의 구현 방법은 입력에 대한 S-Box 출력값을 모두 LUT에 저장하기 때문에 메모리 낭비라는 문제가 있다[4].

그림 5는 SubBytes 연산에서 곱셈역원 연산을 위한 이반적인 S-Box의 구조를 나타낸다.

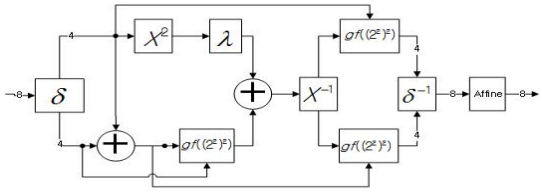

그림 5. 곱셈역원 연산을 위한 S-Box 구조[4]
Fig. 5. S-Box structure for multiplicative inverse[4]

그림 5에서 알 수 있듯이 입력된 8비트의 입력 값은 체 변환 행렬에 의해 $GF(2^8)$에서 $GF((2^2)^2)$ 로 변환 후. 곱셈역원을 계산한다. 이때, 체 역변환 행렬 연산에 의해 다시 $GF(2^8)$의 형태로 변환되어 Affine 연산에 의해 최종 출력을 얻는다. $GF(((2^2)^2)^2)$ 상의 곱셈역원 연산을 위해 $GF((2^2)^2)$ 상의 곱셈 연산기 3개와 $GF(2^4)$ 상의 제곱 및 역원회로, 그리고 XOR 연산회로가 필요하다[5-6].

본 논문에서는 메모리 용량과 연산속도 면에서 보다 효율적인 개선된 S-Box의 구조를 설계한다. 제안한 방법은 $GF(((2^2)^2)^2)$ 상에서 보다 빠른 연산을 위해 x^2과 λ를 1개 모듈로 통합한다. 또한, 체 역변환(δ^{-1}) 및 아핀연산 모듈도 1개로 통합하여 간략화시킨다.

그림 6은 개선된 S-Box 구조를 구현하기 위한 Function getSBOX()를 나타낸다.

```
public byte getSBOX(byte input){
        del = delta(input);
        del_high = (byte)(del 〉〉 4);
        del_low = (byte)(del & 0x0f);
        val = Op_gf8(del_high, del_low);
        output = del_and_aff(val);
        return output;
}
```

그림 6. 개선된 S-Box 구현을 위한 getSBOX() 함수
Fig. 6. getSBOX() Function for advanced S-Box implementation

그림 6에서 알 수 있듯이 체 역변환(δ^{-1}) 및 아핀연산 모듈이 1개의 함수(del_and_aff(val))로 통합하여 연산을 수행하고 있다. 그리고 체 변환 행렬은 delta() 함수에서 연을 하고 있다.

그림 7은 그림 6에 나타난 함수 getSBOX()내에서 합성체 연산을 수행하기 위한 서브루틴인 Op_gf8() 함수를 나타낸다.

```
publicbyte Op_gf8(byte high, byte low) {
    tmp2= lambdaAndSquare ((byte) (high));
    tmp3 = (byte)(high ^ low);
    tmp4 = mul_gf2_4(tmp3, low);
    tmp5 = (byte)(tmp2 ^ tmp4);
    tmp6 = inv_lut(tmp5);
    out_high = mul_gf2_4(high, tmp6);
    out_low = mul_gf2_4(tmp3, tmp6);
    output = (byte)((out_high << 4) | out_low);
    return output;
}
```

그림 7. 서브루틴 Op_gf8()
Fig. 7. subroutine Op_gf8()

그림 7에서 알 수 있듯이 방법은 $GF(((2^2)^2)^2)$ 상에서 보다 빠른 연산을 위해 x^2과 λ를 1개의 함수(lambdaAndSquare())로 통합하여 연산을 수행하고 있다. 함수 mul_gf2_4()는 체 변환 행렬에 의해 $GF((2^2)^2)$ 로 변환한 후에 곱셈역원을 수행한다.

제안한 방법을 사용하면 S-Box 출력값을 LUT와 같은 테이블로 저장하지 않고 입력값에 대한 출력값을 S-Box를 이용하여 연산함으로써 메모리 절약과 속도 향상의 효과를 얻을 수 있다.

그림 8은 critical path 지연의 최소화를 위해 사용된 체 변환과 체 역변환 행렬을 나타낸다[6].

$$\delta = \begin{pmatrix} 11000010 \\ 01001010 \\ 01111001 \\ 01100011 \\ 01110101 \\ 00110101 \\ 01111011 \\ 00000101 \end{pmatrix} \quad \delta^{-1} = \begin{pmatrix} 10101110 \\ 00001100 \\ 01111001 \\ 01111100 \\ 01101110 \\ 01000110 \\ 00100010 \\ 01000111 \end{pmatrix}$$

그림 8. 체 변환 행렬 δ, 체 역변환 행렬δ^{-1}
Fig. 8. Field transformation matrix δ, field inverse transformation matrix δ^{-1}

IV. 시스템 구현 및 성능 평가

본 논문에서 제안한 합성체 S-Box 기반의 CCTV 영상 시스템은 Visual Studio 2010을 사용하여 C# 언어로 구현하였다. 또한, 시스템 구성 환경은 윈도우7 OS 상에서 RAM 8G를 사용하였다.

그림 9는 구현한 프로그램의 실행 결과 중 입력된 평문과 Key 및 각 라운드 별 RoundKey와 암호문을 보여준다.

그림 9에서 알 수 있듯이 128-비트의 평문과 128-비트의 암호화 키를 사용하여 12비트의 암호화된 결과가 출력됨을 보인다. 암호문의 검증을 위해 동일한 암호화 키로 암호문을 복호화한 결과 입력된 평문과 동일한 결과 값을 얻을 수 있음을 확인하였다.

그림 9. 실행 결과
Fig. 9. Execution results

그림 10은 암호화시 기존의 방법[3]과 제안한 방법에 대한 메모리 사용량을 나타내며, 그림 11은 복호화에 사용되는 메모리 사용량을 보인다.

그림 10과 11에서 알 수 있듯이 암호화와 복호화의 경우 메모리 사용량 비교시, 기존의 LUT 기반의 S-Box[3]를 사용하는 방법 보다 본 논문에서 제안한 합성체 S-Box를 사용하여 구현한 방법이 보다 효율적임을 알 수 있다. 표 1은 기존의 S-Box 와 개선된 S-Box에 대한 암·복호화의 속도향상 비율을 나타낸다.

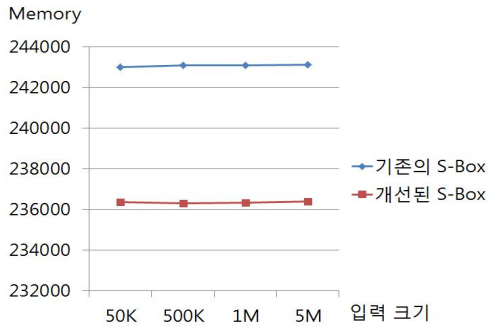

그림 10. 암호화시 메모리 사용량
Fig. 10. Memory usage for encryption

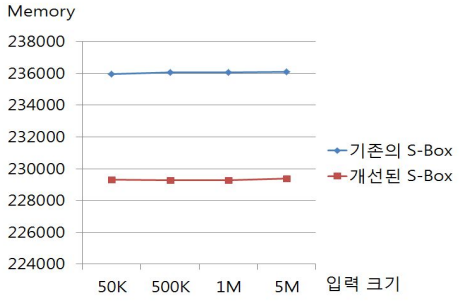

그림 11. 복호화시 메모리 사용량
Fig. 11. Memory usage for decryption

표 1. 제안한 시스템의 암·복호화 속도 향상 비율
Table 1. Speed up rate for encryption and decryption of proposed system

비율 / 입력크기	속도 향상 비율 (LUT 기반 대비)
50K	7.5%
500K	7.6%
1M	8.7%
5M	7.5%

표 1에서 알 수 있듯이 제안한 방법은 암·복호화 시간 비교시 평균 8% 정도의 속도가 향상됨을 확인할 수 있다.

그림 12는 C#을 사용하여 실행된 CCTV 영상 보안 시스템의 구현결과를 보여준다. 시스템의 실험 및 구현 결과를 통하여 차이값 추출, 블록화, 블록 정렬, 암호화, 복호화 과정이 정확하게 동작됨을 확인하였다.

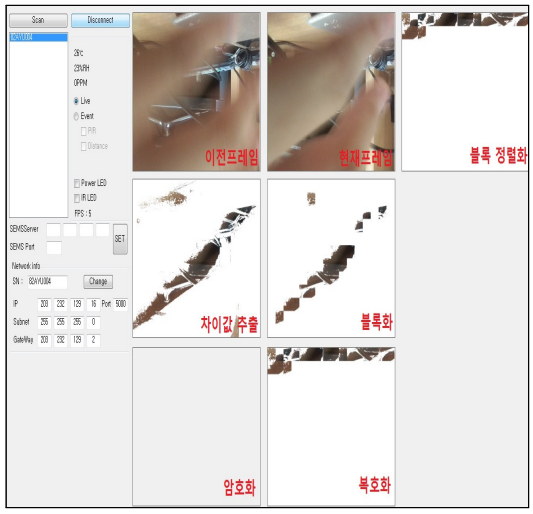

그림 12. 프로그램 실행 결과
Fig. 12. Result of the program execution

V. 결 론

본 논문에서는 합성체 S-Box기반의 CCTV 영상 보안 시스템 구현에 관하여 기술하였다. 제안한 방법에서는 시스템의 효율적인 구현을 위해 AES 알고리듬의 각 라운드 과정에서 요구되는 지연시간의 최소화방안을 제안하였다.

구현된 프로그램은 C#을 사용하여 코딩작업을 행하였으며, CCTV 기기는 Seloco사의 SN200을 사용하였다. 실험 결과를 통하여 기존의 LUT기반의 S-Box와 비교하여 개선된 방법은 메모리 사용량은 약 3% 정도 절약되었고, 속도는 약 8% 정도 향상되었다.

암호화 기술을 적용하여 구현된 영상 보안시스템은 외부에서 해킹을 당하더라도 그 내용을 해독할 수 없기 때문이 기밀성이 보장된다.

References

[1] Min-seok Kang, Ji-su Bae, Tae-min Jang,Min-sup Kang, Design of Digital Image Security System Using AES Algorithm, Journal of Security Engineering, (2011)Vol.8, No.2, pp.277-288.

[2] FIPS 197, "Advanced Encryption Standard (AES)", November 26, 2001.

[3] M. McLoone and J. V. McCanny, "Rijndael FPGA

implementation utilizing look-up tables," in IEEEWorkshop on Signal Processing Systems, pp. 349–360, Sept. 2001.

[4] Edwin NC Mui, "Practical Implementation of Rijndael S-Box Using Combinational Logic", Custom R&D Engineer Texco Enterprise Pvt.Ltd, 2007.

[5] Saurabh Kumar, V.K. Sharma and K. K. Mahapatra, "Low Latency VLSI Architecture of S-Box for AES Encryption" IEEE International Conference on Circuits, Power and Computing Technologies (ICCPCT), March 2013.
https://doi.org/10.1109/ICCPCT.2013.6528906

[6] Su-Bong Ryu, Min-Sup Kang, "Implementation of Image Security System for CCTV Using Analysis Technique of Color Informations", The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 12, No. 5, pp. 219-227, Oct. 2021.
https://doi.org/10.7236/JIWIT.2012.12.5.219

[7] Prathiba, A., Kanchana Bhaaskaran, V.S.: Lightweight S-Box architecture for the secure internet of things. Information 9(1), pp. 1–14, 2018.

[8] Min-Sup Kang, "Design of AES-Based Encryption Chip for IoT Security", The Journal of the Institute of Internet, Broadcasting and Communication, Vol. 21, No. 1, pp. 1-6, Feb. 2021.
https://doi.org/10.7236/JIIBC.2021.21.1.1

[9] Se-Hwan Park, Jong-Kyu Park, "IoT Industry & Security Technology Trends", International journal of advanced smart convergence(IJASC), Vol. 5, No. 3 pp. 27-3, 2016.
https://doi.org/10.7236/IJASC.2016.5.3.27

저 자 소 개

Min-Sup Kang(정회원)

- 1979 : BS degree in Dept of Telecommunication Engineering, Kwangwoon University
- 1984 : MS degree in Dept of Electronic Engineering, Hanyang University
- 1992 : Ph. D. degree in Dept of Electronic Engineering Osaka University
- 1984 ~ 1992 : Senior researcher, ETRI
- 2001 ~ 2002 : Visiting scholar in Dept of electrical & computer Engineering, University of California, Irvine
- 1993 ~ present : Professor, Dept of Computer Engineering, Anyang University
- Research interests : CCTV security, embedded system, crypto-processor design, network security, IOT