SEED 형식 암호에서 공격에 강한 S 박스와 G 함수의 실험적 설계

박 창수* · 송 홍복** · 조 경연*

Experimental Design of S box and G function strong with attacks in SEED-type cipher

Chang-Soo Park* · Hong-Bok Song** · Gyeong-Yeon Cho*

요 약

본 논문에서는 $GF(2^n)$상 곱셈의 복잡도와 규칙도를 GF(2)상의 다항식 곱셈을 표현하는 행렬식의 행과 열의 해밍 가중치를 이용하여 정의한다. 차분공격에 강한 블록 암호 알고리즘을 만들기 위해서는 치환계층과 확산계층의 $GF(2^n)$상 곱셈의 복잡도와 규칙도가 높아야함을 실험을 통하여 보인다. 실험 결과를 활용하여 우리나라 표준인 128 비트 블록 암호 알고리즘인 SEED의 S 박스와 G 함수를 구성하는 방식을 제안한다.

S 박스는 비선형함수와 아핀변환으로 구성한다. 비선형함수는 차분공격과 선형공격에 강한 특성을 가지며, '0'과 '1'을 제외하고 입력과 출력이 같은 고정점과 출력이 입력의 1의 보수가 되는 역고정점을 가지지 않는 $GF(2^8)$ 상의 역수로 구성한다. 아핀변환은 입력과 출력간의 상관을 최저로 하면서 고정점과 역고정점이 없도록 구성한다.

G 함수는 4개의 S 박스 출력을 $GF(2^8)$ 상의 4×4 행렬식을 사용하여 선형변환한다. 선형변환 행렬식 성분은 높은 복잡도와 규칙도를 가지도록 구성한다. 또한 MDS(Maximum Distance Separable) 코드를 생성하고, SAC(Strict Avalanche Criterion)를 만족하고, 고정점과 역고정점 및 출력이 입력의 2의 보수가 되는 약한 입력이 없도록 G 함수를 구성한다.

비선형함수와 아핀변환 및 G 함수의 원시다항식은 각기 다른 것을 사용한다.

본 논문에서 제안한 S 박스와 G 함수는 차분공격과 선형공격에 강하고, 약한 입력이 없으며, 확산 특성이 우수하므로 안전성이 높은 암호 방식의 구성 요소로 활용할 수 있다.

Abstract

In this paper, complexity and regularity of polynomial multiplication over GF(2n) are defined by using Hamming weight of rows and columns of the matrix over GF(2) which represents polynomial multiplication. It is shown experimentally that in order to construct the block cipher robust against differential cryptanalysis, polynomial multiplication of substitution layer and the permutation layer should have high complexity and high regularity. With result of the experiment, a way of constituting S box and G function is suggested in the block cipher whose structure is similar to SEED, which is KOREA standard of 128-bit block cipher.

* 부경대학교 전자컴퓨터정보통신공학부 ** 동의대학교 전자.정보통신공학부

접수일자 : 2003. 1. 15

S box can be formed with a nonlinear function and an affine transform. Nonlinear function must be strong with differential attack and linear attack, and it consists of an inverse number over GF(2^8) which has neither a fixed point, whose input and output are the same except 0 and 1, nor an opposite fixed number, whose output is one's complement of the input. Affine transform can be constituted so that the input/output correlation can be the lowest and there can be no fixed point or opposite fixed point.

G function undergoes linear transform with 4 S-box outputs using the matrix of 4×4 over GF(2^8). The components in the matrix of linear transformation have high complexity and high regularity. Furthermore, G function can be constituted so that MDS(Maximum Distance Separable) code can be formed, SAC(Strict Avalanche Criterion) can be met, and there can be no weak input, where a fixed point, an opposite fixed point, and output can be two's complement of input.

The primitive polynomials of nonlinear function, affine transform and linear transformation are different each other.

The S box and G function suggested in this paper can be used as a constituent of the block cipher with high security, in that they are strong with differential attack and linear attack with no weak input and they are excellent at diffusion.

키워드

SEED, S 박스, 암호, 블록 암호, MDS 코드, SAC, 약한 입력

Ⅰ. 서 론

1977년 IBM이 개발하고, 미국 정부에 의해 수정되어 미국 정부의 암호 표준으로 DES(Data Encryption Standard)[1]가 채택되어 사용된 이후로 많은 암호 방식들이 연구 발표되고 있다. DES는 비선형 치환을 수행하는 S 박스와 S 박스 출력을 교환하는 교환-네트워크로 구성되어 있다. Heys와 Tavares[2-4]는 교환-네트워크를 선형변환으로 바꿈으로써 확산 특성을 개선할 수 있고, 차분공격[5] 및 선형공격[6]에 대한 안전성이 높아진다는 연구 결과를 얻었다. Vaudenay는 선형변환이 MDS(Maximum Distance Separable) 코드를 생성하면 암호 공격에 대한 안전성이 높아진다는 것을 발표하였으며[7], 이러한 연구결과는 SHARK[8]와 SQUARE[9]에 적용되었다. Youssef는 MDS 코드를 생성하는 GF(28) 상에서 4 × 4 선형변환 행렬식은 무작위로 선출하여 생성할 수 있음을 증명하였고, 8 × 8 선형변환 행렬식을 생성하는 알고리즘을 연구하였다[10].

한편 암호 공격 기술에 대한 연구가 발전하여서 DES에 대한 공격이 용이해지자, 미국 국립 표준 기술연구소(NIST, National Institute Standards & Technology)에서는 새로운 표준 블록 암호 알고리즘을 선정하기 위해 1997년 초 새로운 표준 암호 알고리즘(AES, Advanced Encryption Standard)선정 프로젝트를 발표했고, 여러 차례 평가에 의해 5개의 후보 알고리즘(Rijndael, Twofish, MARS, RC6, Serpent)을 선정하였다. 5개의 후보 중 NIST 자체평가와 전 세계적인 공개 검증을 통해 지난 2000년 10월에 최종적으로 Rijndael을 AES로 선정하였다[11,12]. AES는 SHARK 및 SQUARE와 동일한 형태의 치환 블록을 사용하고 있다.

이들 암호 알고리즘에 대한 대부분의 연구는 미국, 유럽 국가 등 몇몇 암호 선진국에서 주도하고 있으며, 정보의 유출을 방지하기 위해서 암호 기술 및 제품에 대한 수출을 규제하고 있다. 이에 자국의 정보를 보호하기 위하여 각 나라들은 자체적인 암호 알고리즘을 연구하고 있으며, 우리나라에서는 학국정보보호센터를 주축으로 관련 전문가들과 공동으로 128 비트 블록 암호 알고리즘인 SEED[13-17]를 개발하여 공개하였다.

SEED는 16 회전을 수행하는 피스탈 네트워크(Feistel network) 구조를 가지며, 64 비트 평문을 2개의 32 비트 블록으로 나누고, 하나의 32 비트 블럭을 G 함수에 의하여 변환하고, 변환한 32 비트와 변환하지 않은 32 비트 블록을 연산하여 32 비트 결과를 구한다. 이러한 과정을 3회 반복하여 하나의 F 함수를 구성한다. G 함수는 4개의 8 비트 S 박스와 S 박스 출력을 선형변환하는 부분으로 구성되어 있는 32 비트 치환 블록이다.

본 논문에서는 비선형 치환을 수행하는 S 박스와 선형변환으로 구성되는 블록 암호알고리즘을 SPS(Substitution Permutation Substitution) 함수

로 모델화하고, SPS 함수의 차분 공격에 대한 안정성을 높이기 위한 연구를 수행한다.

GF(2^n)상 곱셈은 GF(2)상의 다항식 곱셈을 표현하는 행렬식으로 변환하고, 행렬식의 행과 열의 해밍 가중치로 GF(2n)상 곱셈의 복잡도와 규칙도를 정의한다. 실험을 통하여 차분공격에 강한 SPS 함수를 구성하기 위해서는, 비선형 치환계층과 선형변환계층의 원시다항식이 서로 달라야하며, 선형변환 행렬식이 순환이고, 그 인자의 복잡도와 규칙도가 높아야함을 보인다. 이러한 실험결과를 활용하여 우리나라 표준인 128 비트 블록 암호 알고리즘인 SEED의 S 박스와 G 함수를 구성하는 방식을 제안한다.

S 박스는 비선형함수와 아핀변환으로 구성한다. 비선형함수는 차분공격과 선형공격에 강한 GF(2^8)상의 역수[19]를 채택하고, 원시다항식은 '0'과 '1' 이외의 약한 입력을 가지지 않으면서 입출력의 상관계수가 작은 것을 선정한다. 아핀변환은 차분공격과 선형공격의 특성에 영향을 주지 않지만[20], 수식의 복잡도를 증가시켜서 보간공격(interpolation attack)[21]에 강하도록 한다. 본 논문에서는 입력과 출력이 같거나 출력이 입력의 1의 보수가 되는 약한 입력을 가지지 않으면서 입출력의 상관계수가 가장 작은 아핀변환을 구성한다.

SEED G 함수의 선형변환 부분은 간단한 수식으로 되어 있어서 충분한 확산이 이루어지지 않는다. 이러한 단점을 개선하기 위하여 본 논문에서는 F 함수의 특성을 분석하여 G 함수는 전단사함수가 되어야 하며, MDS 코드를 생성하여야 하며, SAC(Strict Avalanche Criterion)[22]를 충족시켜야 함을 제시한다. 또한 G 함수의 약한 입력으로 고정점과 역고정점 및 출력이 입력의 2의 보수가 되는 입력이 약한 입력임을 보인다.

본 논문에서 생성한 G 함수는 차분공격과 선형공격에 강하고, MDS 코드를 생성하고, SAC를 만족하여 확산 특성이 우수하고, 약한 입력을 가지지 않아서 안전성이 높은 암호 방식의 구성 요소로 활용할 수 있다.

II. SPS 함수의 실험적 특성

블록 암호에서 많이 사용하는 SPN(Substitution Permutation Network)의 한 라운드는 치환(substitution), 교환(permutation) 그리고 키 덧셈(key addition)의 3 계층으로 구성된다.

치환계층은 S 박스라고 칭하는 작은 규모의 비선형치환이다. S 박스를 구성하는 방법은 여러 가지가 있는데, 본 논문에서는 '$A \cdot X^{-1} \oplus C$', 단 A, X, C $\in$ GF(2^n)의 형태를 선정한다. GF(2^n)상의 역수 및 아핀 변환에 사용하는 원시다항식을 각각 Ps와 Pa로 정의한다.

교환계층은 GF(2^n)상에서의 선형변환으로 정의할 수 있다. 본 논문에서는 GF(2^n)^m 상의 원소를 GF(2^n)^m 상의 원소로 일대일 선형변환하며, 사용하는 원시다항식을 Pp로 정의한다.

키 덧셈계층은 주 키로부터 생성하는 라운드 키를 암호화 과정에 추가하는 계층으로 정의할 수 있다. 키 덧셈계층은 암호화의 여러 다른 단계에 위치할 수 있는데, 키 덧셈 연산은 주로 배타적 논리합을 사용하고, 이것은 차분공격 특성에 영향을 주지 않으므로 안정성 분석 과정에서 종종 생략한다. 산술 덧셈을 사용하는 경우에도 배타적 논리합으로 근사할 수 있으므로 안전성 분석 과정에서 생략할 수 있다. 치환계층의 아핀변환에서 상수 C를 더하는 과정도 동일한 이유에서 생략할 수 있다.

SPN 한 라운드에서 키 덧셈 계층을 생략하면 치환-교환 계층으로 취급할 수 있다. 두 라운드 치환-교환 계층을 하나로 묶으면 치환-교환-치환-교환 계층이 되는데, 마지막 교환계층은 선형변환이므로 차분 및 선형 공격 특성에 영향을 주지 않는다. 따라서 두 라운드의 SPN은 하나의 SPS(Substitution Permutation Substitution) 함수로 취급할 수 있다. 본 논문에서는 블록 크기 N, S 박스의 수 m, S 박스의 길이 n으로 SPS 함수를 정의한다. 그림-1에 N=15, n=5, m=3 인 SPS 함수를 보인다.

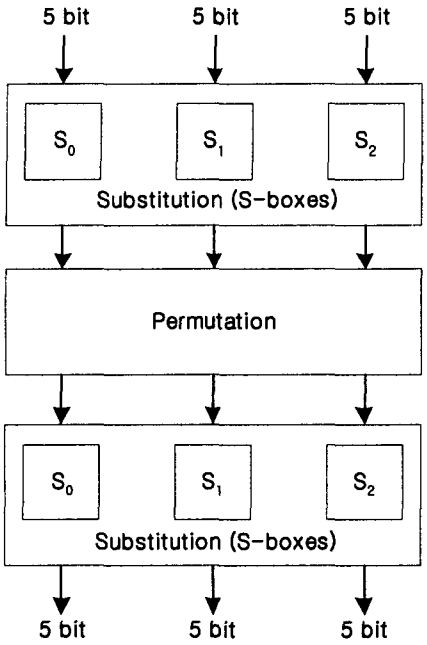


그림 4. N=15, n=5, m=3 SPS 함수의 블록도
Fig. 1. SPS function block diagram with N=15, n=5 and m=3

본 논문에서는 S 박스는 전단사함수로 구성되어 있고, 교환계층의 선형변환은 MDS(Maximum Distance Separable) 코드를 생성한다고 가정한다.

1. GF(2^n)상의 곱셈 특성

치환 및 교환 계층은 GF(2^n)상의 곱셈을 포함한다.

Lemma-1) GF(2^n) 상의 곱셈 'C = A × B'은 B가 상수라면 GF(2)상에서의 행렬식 곱셈 $|C|^T = |M| \times |A|^T$ 가 된다.

증명) GF(2^n) 상의 다항식 A와 B는 각각 $A = \sum_{i=0}^{n-1} a_i x^i$, $B = \sum_{i=0}^{n-1} b_i x^i$ 로 표현할 수 있다. 또한 원시다항식 P는 $P = \sum_{i=0}^{n} p_i x^i$ 로 표현할 수 있다.

단위 함수 u()를 다음과 같이 정의하면,

$$u(i, j, m) = 1 \text{ if } m = i + j$$
$$= 0 \text{ if } m \neq i + j$$

다항식 곱셈 'C = A × B'는 다음과 같이 된다.

$$C = A \times B = \sum_{i=0}^{2n-2} c_i x^i$$

$$c_m = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} u(i, j, m) \cdot a_i \cdot b_j,$$
$$\text{while } m = (0, 1, \cdots, 2n-2)$$

C 다항식의 계수는 다음과 같은 과정으로 계산할 수 있다.

```
for (i=2n-2 ; i > n-1 ; i--)
    for (j=0 ; j < n+1 ; j++)
        c_{i-j} = c_i · p_{n-j} ⊕ c_{i-j}
```

따라서 $c_i \in \{c_{n-1}, c_{n-2}, \cdots, c_0\}$은

$$c_i = \sum_{j=0}^{n-1} f_j(b_{n-1}, b_{n-2}, \cdots, b_0, p_{n-1}, \cdots, p_0) \cdot a_j,$$
$$\text{where } b_i, p_i, f_j(\ldots) \in GF(2)$$

이 된다. □

정의-1) Lemma-1에서 정의한 GF(2)상의 n × n 행렬식에서 i 행의 원소 '1'의 개수, 즉 해밍 가중치를 $W_{row,i}$으로 표현하고, j 열의 해밍 가중치를 $W_{col,j}$으로 표현하면, GF(2^n)상 곱셈의 복잡도 CM과 규칙도 RM을 각각 다음과 같이 정의한다.

$$CM = \min(W_{row,i}, W_{col,j}),$$
$$\text{while } i, j=(0,1, \cdots, n-1)$$
$$RM = CM/\max(W_{row,i}, W_{col,j}),$$
$$\text{while } i, j=(0,1, \cdots, n-1)$$

2. 교환 계층의 특성

N 비트의 입력과 출력을 가지는 SPS 함수를 H라고 하면, H : $GF(2^n)^m \rightarrow GF(2^n)^m$, while N = nm 으로 표현할 수 있다.

정의-2) H의 최대 차분 확률(differential probabilities) MP는 다음과 같이 정의된다.

$$DP(\Delta X \rightarrow \Delta Y) = \frac{Numberof\,[(H(X) \oplus H(X+\Delta X)) = \Delta Y]}{2^{nm}},$$

while $\{X, \Delta X, \Delta Y\} \in GF(2^n)^m$

$MP = \max\{DP(\Delta X \rightarrow \Delta Y)\}$, while $\Delta X \neq 0$

정의-3) $D_0, D_1, \cdots, D_k$를 교환 계층의 선형변환 행렬이라고 하면, H의 최적(lowest) 차분 확률 bMP와 최악(highest) 차분 확률 wMP를 각각 다음과 같이 정의한다.

$bMP = \min MP$, while H with $D_0, D_1, \cdots, D_k$

$wMP = \max MP$, while H with $D_0, D_1, \cdots, D_k$

정의-4) H의 규칙도 RP를 다음과 같이 정의한다. RP가 1에 가까우면 차분공격 특성은 교환 계층의 선형변환 행렬식과 무관함을 나타낸다.

$$RP = bMP/wMP$$

3. SPS 함수 모델

SPS 함수의 이론적인 차분 및 선형 공격에 대한 확률은 Heys[4] 등의 연구 결과에 나와있다. 실제적인 공격에 대한 안정성은 이들 논문에 근거하여 추론하고 있다. 그러나, 실제적인 확률은 이론적인 확률 값과 큰 차이를 보이고 있다. 본 논문에서 다음과 같은 3가지 모델의 SPS 함수에 대하여 실제적인 차분공격에 대한 확률을 구하기 위하여 소모적 실험(exhaustive experiment)를 수행한다.

model-1 : N=16, n=8, m=2
model-2 : N=15, n=5, m=3
model-3 : N=20, n=5, m=4

4. 실험 결과

SPS 함수 모델에 대한 소모적 실험 결과는 다음과 같다.

결과-1) 치환계층의 S 박스와 아핀 변환 및 교환계층에 모두 동일한 원시 다항식을 사용하면, 즉, Ps=Pa=Pp 이면, H의 최악 차분 확률 wMP가 상당히 높게 나타난다. 이것은 치환계층의 곱 다항식이나 교환계층의 선형변환 행렬식 인자에 상관없이 이러한 현상을 보인다. SPS 모델-1에서는 $wMP=256\cdot2^{-16}$을 보였고, 모델-2와 모델-3에서는 각각 $64\cdot2^{-15}$ 과 $256\cdot2^{-20}$을 보였다.

이 결과로부터 Ps=Pa=Pp가 되는 DONUT[5]는 차분공격에 약하다는 것을 알 수 있다.

결과-2) 치환계층의 S 박스와 교환계층의 원시다항식은 동일하지만 치환계층 아핀변환의 원시다항식이 다르고, 즉, Ps=Pp≠Pa 이고, 아핀변환의 곱셈 다항식의 복잡도 CM 및 규칙도 RM이 낮으면, 교환계층의 선형변환 행렬식에 관계없이 H의 최악 차분 확률 wMP가 높게 나타난다. SPS 모델-1에서 $wMP=80\cdot2^{-16}$을 보였다.

이 결과로부터 AES[11]는 Ps=Pp이지만 아핀변환의 곱셈 다항식의 복잡도 및 규칙도 RM이 높아서 차분공격에 강하다는 것을 알 수 있다.

결과-3) 치환계층의 S 박스와 아핀변환 및 교환계층에 각기 다른 원시다항식을 사용하면, 즉, Ps≠Pa, Pa≠Pp, Ps≠Pp 이면, 아핀변환의 곱 다항식의 복잡도 CM 및 규칙도 RM는 H의 차분공격 특성에 영향을 주지 않는다.

결과-4) 교환계층의 선형변환 행렬식이 순환(circular)이고 그 인자들의 복잡도 CM 및 규칙도 RM이 높으면, H의 규칙도 RP가 높게 나타난다. SPS 모델-1의 높은 규칙도의 선형변환 행

렬식에서 bMP=20·2^{-16}, wMP=24·2^{-16}를 나타냈지만, 낮은 규칙도에서는 wMP=32·2^{-16}가 되었다. 모델-2와 모델-3에서도 동일한 실험 결과를 보이고 있다. 순환 행렬식이 아니면 모델-1에서 wMP=40·2^{-15}으로 높게 나타난다.

본 실험 결과로부터 차분공격에 강한 특성을 보이기 위해서는 교환계층의 선형변환 행렬식은 순환이고 그 인자들의 복잡도 CM 및 규칙도 RM이 높아야 한다.

III. SEED S 박스 및 G 함수 설계

SPS 함수의 실험적 결과를 우리나라 블록 암호 알고리즘의 표준인 SEED에 적용해서 S 박스와 G 함수를 설계한다.

1. S 박스의 설계

차분공격에 강한 S 박스를 구성하기 위해서는 차분 XOR 테이블에서 모든 첫 번째 컬럼이 '0'이 되어야 하며, '0'인 성분의 수가 작아야 하며, 최대 성분 값이 작아야 한다[23-25]. 첫 번째 조건을 만족시키기 위해서는 S 박스는 전단사함수가 되어야 한다. 두 번째 조건을 만족하기 위해서 '0' 또는 '2'인 성분이 많아야 하며, 세 번째 조건을 만족하기 위해서 최대 성분 값이 '4'인 전단사함수를 선정해야 한다. 한편 선형공격에 강한 S 박스를 구성하기 위해서는 입력과 출력의 상관계수가 작아야 한다.

또한 S 박스는 S(A)=A인 고정점과 S(A)=~A인 역고정점이 없어야 한다. 고정점과 역고정점은 S 박스가 치환 기능을 수행하지 않는 입력으로 약한 입력이다.

이들 조건을 만족하는 비선형 전단사함수를 찾기 위해서 GF(2^8)상에서 nl(X) ==> X^{-1}을 계산하였다. 단, 입력 '0'과 '1'은 예외적인 경우로 고정점 판단에서 제외한다. 조건을 만족하는 역수의 계산 결과 일부를 표-1에 보인다.

표 1. 고정점과 역고정점을 가지지 않는 GF(2^8)상의 역수 특성
Table 1. Reciprocal characteristics of none fixed and opposite fixed point on GF(2^8)

원시다항식	입출력 상관계수
0x1a3	0.010
0x163	0.015
0x18b	0.019
0x12b	0.020
0x171	0.037
0x1dd	0.047
0x19f	0.052
0x1c3	0.066
0x1cf	0.080
0x15f	0.080
0x1a9	0.081
0x165	0.111

표-1은 입출력 상관계수가 작은 순서대로 나열하였다.

S 박스는 비선형함수와 아핀변환을 결합하여 식-1과 같이 표현된다.

$$S(X) = ((X^{-1} \bmod Q) \times M) \bmod P + C \quad (1)$$

o M과 P는 GF(2^8)상에서 서로소(relatively prime)

실험 결과-3으로부터 식-1에서 비선형함수와 아핀 변환의 원시다항식은 상호 다르게 선정한다. 아핀변환은 차분공격과 선형공격 특성에 변화를 주지 않지만 수식의 복잡도를 증가시켜서 보간 공격에 강한 특성을 나타내며, 비선형함수의 고정점을 없애는 기능을 수행한다.

식-1에서 고정점과 역고정점을 가지지 않으면서 상관계수를 최소로 하는 아핀함수를 구하여 정리한 것을 표-2에 보인다.

2. G 함수의 특성

SEED의 G 함수는 32 비트 입력 X를 32 비트 출력 Z로 변환하는 치환 블록으로 G : X → Z, {X, Z} ∈ $GF(2^8)^4$ 함수이며 식-2와 같이 정의할 수 있다. 입출력 관계식을 복잡하게 하여서 안전성을 높이기 위해서 $GF(2^8)$상의 서로 다른 4개의 S 박스, S_0-S_3를 사용한다.

$$\begin{vmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \end{vmatrix} = \begin{vmatrix} A_{00} & A_{01} & A_{02} & A_{03} \\ A_{10} & A_{11} & A_{12} & A_{13} \\ A_{20} & A_{21} & A_{22} & A_{23} \\ A_{30} & A_{31} & A_{32} & A_{33} \end{vmatrix} \cdot \begin{vmatrix} S_0(X_0) \\ S_1(X_1) \\ S_2(X_2) \\ S_3(X_3) \end{vmatrix} \quad (2)$$

식-2에서 G 함수는 S 박스와 $GF(2^8)$ 상의 4 X 4 행렬식으로 표현되는 선형변환 부분의 결합으로 표현할 수 있다. 32 비트 입력 X는 4개의 8 비트 $X_i \in \{X_3, X_2, X_1, X_0\}$로 분할되어 $S_i \in \{S_3, S_2, S_1, S_0\}$ 박스에서 치환되고, 선형변환 행렬식에서 변환되어 4개의 8 비트 $Z_i \in \{Z_3, Z_2, Z_1, Z_0\}$로 출력된다. X_3와 Z_3가 최상위 바이트이다.

2-1. 전단사함수

SEED의 F 함수를 변수 T0와 T1에 아래바첨자를 붙여서 변화하는 모습을 알기 쉽도록 표현하면 다음과 같이 된다.

```
T0_0 = R0 ^ roundkey[Ki][0] ;
T1_0 = R1 ^ roundkey[Ki][1] ;

T1_1 = T0_0 ^ T1_0 ;
T1_2 = G(T1_1) ;
T0_1 = T0_0 + T1_2 ;
T0_2 = G(T0_1) ;
T1_3 = T0_2 + T1_2 ;
T1_4 = G(T1_3) ;
T0_3 = T0_2 + T1_4 ;

R0' = T0_3;
```

표 2. S 박스 계수표

Table 2. S box table

비선형함수 원시다항식 Q	아핀변환 곱항 M	아핀변환 원시다항식 P	아핀변환 상수 C	S 박스 입출력 상관계수	비고
0x1a3	0x38	0x1c5	0x94	0.000056	G0-S0 박스
0x163	0xc8	0x159	0x3f	0.000303	G0-S1 박스
0x18b	0xba	0x1cd	0x11	0.000112	G0-S2 박스
0x12b	0x71	0x182	0xd9	0.000296	G0-S3 박스
0x171	0xc1	0x1a4	0x40	0.000411	G1-S0 박스
0x1dd	0x4d	0x1fc	0x2f	0.000149	G1-S1 박스
0x19f	0x84	0x17b	0x09	0.000095	G1-S2 박스
0x1c3	0xf4	0x1e1	0x08	0.000130	G1-S3 박스
0x1cf	0x76	0x127	0x63	0.000400	G2-S0 박스
0x15f	0x44	0x16f	0x2c	0.000202	G2-S1 박스
0x1a9	0x43	0x1bf	0x2e	0.000518	G2-S2 박스
0x165	0x06	0x11b	0x18	0.000450	G2-S3 박스

R1' = T1_4;

G 함수의 출력 Z의 상태수 N_z가 '$N_z < 2^{32}$'이면, T1_2와 T0_2의 상태수가 N_z가 되며, 이어서 T1_3와 T1_4 및 T0_3의 상태수가 2^{32}보다 작아진다. 이러한 현상을 방지하기 위해서는 G 함수의 Z 출력의 상태수는 반드시 2^{32}가 되어야 한다.

한편 G 함수는 X 입력을 Z 출력으로 변환하는 기능을 수행하는 함수로 차분 공격에 강하기 위해서 단사함수가 되어야 한다. 따라서 G 함수는 전단사함수가 되어야 한다.

2-2. MDS 코드

G 함수를 나타내는 식-2의 선형변환 행렬식은 차분공격과 선형공격에 강하도록 설계되어야 한다. 이를 위해서 차분확산계수(differential branch number)와 선형확산계수(linear branch number)가 모두 최고 값을 가지도록 설계해야 한다.

식-2의 G 함수에서 S 박스의 8 비트 단위 출력을 $P_i \in \{P_0, P_1, P_2, P_3\}$라 하면, 선형변환 행렬식은 Pi를 확산시켜서 출력 $Z_i \in \{Z_0, Z_1, Z_2, Z_3\}$를 생성한다. Wh(a)를 GF($2^8$) 상의 a의 해밍가중치라고 하면,

Wh(a) = 0 if a = 0
= 1 if a ≠ 0

이 된다.

선형변형 행렬식의 차분확산계수 Bd는 식-3과 같이 정의할 수 있다

$$Bd = \min\left(\sum_{i=0}^{3} Wh(P_i) + \sum_{i=0}^{3} Wh(Z_i) \right), \quad \text{while } P \neq 0 \qquad (3)$$

식-3에서 $\sum Wh(P_i)$와 $\sum Wh(Z_i)$의 최대 값은 4이고 최소 값은 1이 된다. 따라서 차분확산계수는 Bd≤5가 된다. 차분확산계수는 입력 P_i가 변화하였을 때 출력 Z_i가 변화하는 GF(2^8) 상의 코드 수의 최소 값을 나타낸다. 차분 공격에 있어서 비활성 S 박스 입력의 차분은 '0'이며, 따라서 출력의 차분도 '0'이다. SEED의 F 함수는 3개의 G 함수를 직렬로 연결한 구조이다. 따라서 차분공격에 강하기 위해서는 G 함수의 차분확산계수는 최고 값인 5가 되어야 한다.

한편 선형변형 행렬식의 선형확산계수 Bl는 식-4와 같이 정의할 수 있다

$$Bl = \min\left(\sum_{i=0}^{3} Wh(Ai) + \sum_{i=0}^{3} Wh(Bi) \right), \quad \text{while } B \neq 0 \qquad (4)$$

식-4에서 A_i와 B_i는 각각 입력 마스크 값과 출력 마스크 값을 나타내며, 식-2의 선형변환 행렬식을 M이라고 하면 $A = M^t \times B$가 된다[25]. 식-4에서 $\sum Wh(A_i)$와 $\sum Wh(B_i)$의 최대 값은 4이고 최소 값은 1이 된다. 따라서 선형확산계수는 Bl≤5가 된다. 선형확산계수는 출력 Z_i에 대하여 선형 결합된 입력 P_i의 GF(2^8) 상의 코드 수의 최소 값을 나타낸다. SEED의 F 함수는 3개의 G 함수를 직렬로 연결한 구조이다. 따라서 선형공격에 강하기 위해서는 G 함수의 선형확산계수는 최고 값인 5가 되어야 한다. 확산계수가 최고 값을 가지는 코드를 MDS (Maximum Distance Separable) 코드라고 한다. G 함수는 MDS 코드를 생성해야 한다.

식-3과 식-4로부터 선형변환 행렬식 M이 대칭 행렬이면 'Bd = Bl'이 된다[27]. 본 논문에서는 대칭 행렬을 구성하기 위하여 한 행의 성분 4개를 선정하고, 아래 행은 윗 행의 성분을 왼쪽으로 회전시켜서 순환행렬을 구성한다. 이렇게 구성한 G 함수를 식-5에 보인다.

$$\begin{vmatrix} Z_0 \\ Z_1 \\ Z_2 \\ Z_3 \end{vmatrix} = \begin{vmatrix} A_0 & A_1 & A_2 & A_3 \\ A_1 & A_2 & A_3 & A_0 \\ A_2 & A_3 & A_0 & A_1 \\ A_3 & A_0 & A_1 & A_2 \end{vmatrix} \cdot \begin{vmatrix} S_0(X_0) \\ S_1(X_1) \\ S_2(X_2) \\ S_3(X_3) \end{vmatrix} \qquad (5)$$

2-3. SAC

식-2의 G 함수에서 두 개의 입력 X와 X_i에 대하여, X_i는 i($0 \le i \le 31$) 비트만이 X와 다르다

표 3. G 함수의 선형변환 행렬식 생성 알고리즘
Table 3. Linear transformation matrix generation algorithm of G function

step-1) 복잡도가 4이고 규칙도가 0.8인 GF(2^8) 상의 행렬식 성분 4개를 무작위로 발췌하여 선형변환 행렬식을 구성한다.
step-2) 행렬식의 역행렬식을 구한다. 역행렬식이 구해지지 않으면 전단사함수가 되지 않으므로 step-1로 되돌아간다.
step-3) 역행렬식의 모든 성분이 '0'이 아닌 것을 확인한다. '0'인 성분이 있으면 확산이 제대로 이루어지지 않으므로 step-1로 되돌아간다.
step-4) 선형변환 행렬식이 입력 $P \in \{1, 2, \cdots, 2^{32}-1\}$에 대하여 식-3의 차분확산계수가 5인가를 판단한다. 5 미만이면 MDS 코드를 생성하지 않으므로 step-1으로 되돌아간다.
step-5) $X \in \{1, 2, \cdots, 2^{32}-1\}$에 대하여 고정점 'G(X)=X'와 역고정점 'G(X)=~X' 및 'G(X)=-X'가 되는 약한 입력점을 가지지 않는 것을 확인한다. 약한 입력점을 가지면 step-1로 되돌아간다.

고 할 때, V_{ij}는 Z의 $j(0 \le j \le 31)$ 비트가 변경될 확률이라고 정의하면, 입력 $X_i \in \{0, 1, \cdots, 2^{31}-1\}$에서 V_{ij}가 평균값 0.5를 가지는 정규분포를 이루면 SAC를 만족시킨다고 한다.

선형공격 및 차분공격에 강하기 위해서는 입력의 작은 비트의 변화가 출력의 많은 비트에 나타나야 하며, 또한 입력의 많은 비트의 변화가 출력의 적은 비트의 변화로 나타나야 한다.

표 2. S 박스 계수표
Table 2. S box table

비선형함수 원시다항식 Q	아핀변환 곱항 M	아핀변환 원시다항식 P	아핀변환 상수 C	S 박스 입출력 상관계수	비고
0x1a3	0x38	0x1c5	0x94	0.000056	G0-S0 박스
0x163	0xc8	0x159	0x3f	0.000303	G0-S1 박스
0x18b	0xba	0x1cd	0x11	0.000112	G0-S2 박스
0x12b	0x71	0x182	0xd9	0.000296	G0-S3 박스
0x171	0xc1	0x1a4	0x40	0.000411	G1-S0 박스
0x1dd	0x4d	0x1fc	0x2f	0.000149	G1-S1 박스
0x19f	0x84	0x17b	0x09	0.000095	G1-S2 박스
0x1c3	0xf4	0x1e1	0x08	0.000130	G1-S3 박스
0x1cf	0x76	0x127	0x63	0.000400	G2-S0 박스
0x15f	0x44	0x16f	0x2c	0.000202	G2-S1 박스
0x1a9	0x43	0x1bf	0x2e	0.000518	G2-S2 박스
0x165	0x06	0x11b	0x18	0.000450	G2-S3 박스

SEED의 F 함수는 G 함수의 출력이 다음 G 함수의 입력이 되는 구조이다. 또한 입력은 3개의 G 함수를 거쳐서 출력이 된다. 이와 같은 구조에서 G 함수가 SAC를 만족하지 않으면 입력의 변화가 일부 출력에만 나타나고, 이러한 과정이 3번 반복되면 출력의 특정 부분에만 변화가 집중될 수 있다.

SEED에서는 G 함수 출력에 대하여 덧셈 연산을 하여서, 캐리 전파에 의하여 확산을 시키고 있다. 그런데 캐리 전파는 더하는 두 개의 비트가 '0'과 '1'인 경우에만 발생하므로, 캐리 전파 확률은 50%이다. 즉, 덧셈 연산의 캐리 전파만으로는 충분한 확산을 기대할 수 없다.

따라서 입력의 변화를 출력에 충분히 확산시키기 위해서 G 함수는 SAC를 충족시켜야 한다.

2-4. 약한 입력

G 함수는 X 입력을 Z 출력으로 변환하는 함수로 32 비트 치환 블럭이다. 따라서 'G(X) = X'인 고정점과 'G(X)=~X'인 역고정점은 약한 입력이 된다.

또한 F 함수는 G 함수의 출력에 대하여 덧셈 연산을 수행한다. 'G(X) = -X'가 되는 입력 X에 대하여서 F 함수는

```
T0_0 = R0 ^ roundkey[Ki][0] ;
T1_0 = R1 ^ roundkey[Ki][1] ;

T1_1 = T0_0 ^ T1_0 ;
T1_2 = G(T1_1) ;
T0_1 = T0_0 + T1_2 ;
T0_2 = G(T0_1) = -T0_1
     = -T0_0 - T1_2 ;
T1_3 = T0_2 + T1_2 = -T0_0 ;
T1_4 = G(T1_3) = T0_0 ;
T0_3 = T0_2 + T1_4
     = -T0_0 - T1_2 + T0_0 = -T1_2 ;

R0' = T0_3 = -(T0_0 ^ T1_0) ;
R1' = T1_4 = T0_0 ;
```

가 된다. 따라서 'G(X) = -X'가 되는 입력은 약한 입력이다.

G 함수가 약한 입력을 가지면, 약한 입력에 대해서는 치환 기능을 수행하지 않는다. 따라서 G

표 4. G 함수의 S 박스와 선형변환 행렬식
Table 4. Linear transformation matrix of S box on G function

	S 박스	선형변환 행렬식
G0	S_0 = {0x1a3, 0x38, 0x1c5, 0x94} S_1 = {0x163, 0xc8, 0x159, 0x3f} S_2 = {0x18b, 0xba, 0x1cd, 0x11} S_3 = {0x12b, 0x71, 0x182, 0xd9}	A_i = {0x2e, 0x17, 0x5c, 0xb8} Ap = 0x1e7
G1	S_0 = {0x171, 0xc1, 0x1a4, 0x40} S_1 = {0x1dd, 0x4d, 0x1fc, 0x2f} S_2 = {0x19f, 0x84, 0x17b, 0x09} S_3 = {0x1c3, 0xf4, 0x1e1, 0x08}	A_i = {0x17, 0x5c, 0xb8, 0x2e} Ap = 0x1e7
G2	S_0 = {0x1cf, 0x76, 0x127, 0x63} S_1 = {0x15f, 0x44, 0x16f, 0x2c} S_2 = {0x1a9, 0x43, 0x1bf, 0x2e} S_3 = {0x165, 0x06, 0x11b, 0x18}	A_i = {0x17, 0x5c, 0x5c, 0xb8} Ap = 0x1e7

표 5. G 함수 특성표
Table 5. Characteristics of G function

	G0	G1	G2
No of 2 input EXOR gates	528 gates	528 gates	528 gates
Maximum Tpd	5 gates	5 gates	5 gates
SACavr	0.507	0.501	0.505
SACdev	0.032	0.031	0.031
V_{ij}(SACavr±1 SACdev)	60 %	60 %	65 %
V_{ij}(SACavr±2 SACdev)	98 %	97 %	98 %
V_{ij}(SACavr±3 SACdev)	100 %	100 %	100 %

함수는 약한 입력을 가지지 않아야 한다.

3. G 함수 설계

실험결과-4로부터 G 함수의 선형변환 행렬식 인자는 복잡도와 규칙도가 높아야 차분공격에 강한 것을 알 수 있다. GF(2^8)상에서 곱셈의 복잡도의 최대 값은 4이고, 규칙도의 최대 값은 0.8이다. 이러한 조건의 선형변환 행렬식을 표-3의 알고리즘에 의하여 생성한다.

표-3의 알고리즘을 적용하여 생성한 G 함수를 표-4에 보인다.

표-4에서 S_n={Q, M, P, C}는 식-1의 Q, M, P 및 C를 각각 나타낸다.

A_i={A_0, A_1, A_2, A_3}는 식-5의 GF(2^8)상의 4 × 4 순환 행렬식의 각 성분을 나타내며, Ap는 행렬식의 원시다항식이다.

IV. 구현 및 평가

G 함수를 소프트웨어로 구현하는 경우에는 8 × 32 SS 박스를 만드는 것이 효율적이다. SS 박스는 4개의 8 × 32 메모리를 필요로 한다.

하드웨어로 구현하는 경우에는 4개의 8 × 8 S 박스 ROM[14]을 사용하는 경우와 4개의 8 × 32 SS 박스 ROM[17]을 사용하는 경우가 있다. 8 × 8 S 박스 ROM으로 구현하는 경우에 선형변환 행렬식을 구현하는 데 소요되는 2 입력 XOR 게이트의 수 및 최대 전달지연시간을 표-5에 보인다.

확산 특성은 MDS 코드이면서 SAC를 만족하는 가로 판단할 수 있다. 본 논문의 G 함수는 MDS 코드를 생성하므로 SAC를 만족하는 가만을 판단하면 된다. V_{ij}의 평균 SACavr, V_{ij}의 표준편차 SACdev를 산출하여 표-5에 보인다. 표-5에서 V_{ij}(SACavr±n SACdev) 항목은 SACavr의 위, 아래 n배의 SACdev 구간에 분포하는 V_{ij}를 나타낸다. 모든 V_{ij}가 평균값 주위에 밀집한 정규분포를 이루므로 본 논문의 G 함수는 SAC를 만족시킨다.

한편 SEED G 함수의 차분확산계수 Bd는 4로 MDC 코드를 생성하지 않으며, SACavr과 SACdev는 각각 0.376과 0.219로 SAC를 만족시키지 못한다. 또한 SEED의 S1 박스, S2 박스 및 G 함수는 다음과 같은 약한 입력을 가진다.

SEED S1 박스의 약한 입력 :

고정점 : S1(0x17) = 0x17, S1(0xe6) = 0xe6

역고정점 : S1(0x48) = 0xb7, S1(0xfa) =

0x05

SEED S2 박스의 약한 입력 :

고정점 : S2(0x1c) = 0x1c

SEED G 함수의 약한 입력 :

고정점 : G(0x32f732f7) = 0x32f732f7

G(0xa867a867) = 0xa867a867

역고정점 : G(0x5ae96fb0) = 0xa516904f

G(0x6fb05ae9) = 0x904fa516

2의 보수 : G(0xd60b3903) = 0x29f4c6fd

V. 결 론

컴퓨터와 정보통신이 발달할수록 더욱 많은 정보를 처리하게 된다. 이에 따라서 정보에 대한 보안이 중요한 문제로 대두되면서 정보를 보호하고 불법적인 유출을 방지하기 위해서 암호의 필요성이 증대되고 있다. 이러한 필요성에 따라서 우리나라에서도 학국정보보호센터를 주축으로 관련 전문가들과 공동으로 128 비트 블록 암호 알고리즘인 SEED를 1998년에 개발하여 공개하였다.

본 논문에서는 블록 암호에서 많이 사용하는 SPN(Substitution Permutation Network)의 한 라운드를 안정성 분석을 위하여 치환(substitution), 교환(permutation)의 두 개 계층으로 간소화하고, 두 라운드의 SPN을 하나의 SPS (Substitution Permutation Substitution) 함수로 묶어서 표현하였고, SPS 함수의 특성을 실험을 통하여 분석하였다.

치환계층과 교환계층은 $GF(2^n)$상의 곱셈을 포함하는데, $GF(2^n)$상 곱셈을 GF(2)상의 다항식 곱셈을 표현하는 행렬식으로 변환하였고, 행렬식의 행과 열의 해밍 가중치로 $GF(2^n)$상 곱셈의 복잡도와 규칙도를 정의하였다. 실험을 통하여 차분공격에 강한 SPS 함수를 구성하기 위해서는, 비선형 치환계층과 선형변환계층의 원시다항식이 서로 달라야하며, 선형변환 행렬식이 순환이고, 그 인자의 복잡도와 규칙도가 높아야함을 보였다. 이러한 실험 결과를 활용하여 우리나라 표준인 128 비트 블록 암호 알고리즘인 SEED의 S 박스와 G 함수를 구성하는 방식을 제안하였다.

본 논문에서 제안한 S 박스와 G 함수는 차분공격과 선형공격에 강하고, 약한 입력이 없으며, 확산 특성이 우수하므로 안전성이 높은 암호 방식의 구성 요소로 활용할 수 있다.

참고 문헌

[1] ANSI X3.92, "American National Standard for Data Encryption Algorithm(DEA)," NIST, 1983

[2] H.M. Heys and S.E. Tavares, "The Design of Substitution Permutation Networks Resistant to Differential and Linear Cryptanalysis," Proceedings of 2nd ACM Conference on Computer and Communications Security, Fairfax, Virginia, pp. 148-155, 1994

[3] H.M. Heys and S.E. Tavares, "The Design of Product Ciphers Resistant to Differential and Linear Cryptanalysis," Journal of Cryptology, Vol. 9, no. 1, pp. 1-19, 1996

[4] H.M. Heys and S.E. Tavares, "Avalanche Characteristics of Substitution Permutation Encryption Networks," IEEE Transaction on Computer, Vol. 44, pp. 1131-1139, Sep. 1995

[5] E. Biham and A. Shamir, "Differential cryptoanalysis of DES-like cryptosystems," Journal of Cryptology, Vol. 4, No. 1, pp. 3-72, 1991

[6] M. Matsui, "The first experimental cryptanalysis of the Data Encryption Standard," Advances in Cryptology, Proceedings of CRYPTO '94, Springer-Verlag, Berlin, pp. 1-11, 1994

[7] S. Vaudenay, "On the need for multipermutations: Cryptanalysis of MD4 and SAFER," Proceedings of Fast Software Encryption (2), LNCS 1008, Springer-Verlag, pp. 286-297, 1995

[8] V. Rijmen, J. Daemen, B. Preneel, A. Bosselaers and E. De Win, "The cipher SHARK," Fast Software Encryption, LNCS 1039, D. Gollmann, Ed., Springer-Verlag, pp. 99-112, 1996

[9] J. Daemen, L. Knudsen and V. Rijmen,

"The block cipher SQUARE," Proceedings of Fast Software Encryption (4), LNCS, Springer-Verlag, 1997

[10] A.M. Youssef, S. Mister and S.E. Tavares, "On the Design of Linear Transformations for Substitution Permutation Encryption Networks," ACM Symposium on Applied Computing (SAC'97), Feb. 1997

[11] NIST, "Advanced Encryption Standard Development Effort." http://csrc.nist.gov/encryption/aes.

[12] Joan Daemen, Vincent Rijmen, "AES Proposal: Rijndael", 1999

[13] 한국정보보호센터, 128 비트 블록 암호알고리즘(SEED) 개발 및 분석 보고서, Dec. 1998

[14] Young-Ho Seo, Jong-Hyeon Kim and Dong-Wook Kim, "Hardware Implementation of 128-bit Symmetric Cipher SEED," The Second IEEE Asia Pacific Conference on ASICs, pp. 183-186, Aug. 2000

[15] 이 명동, "SEED 암호 알고리즘의 FPGA 구현을 위한 RTL 수준 VHDL 설계," 한남대학교 대학원 컴퓨터공학과 석사학위논문, 2001

[16] 정 찬호, "SEED에 대한 효과적인 Brute-Force 공격 알고리즘," 한국항공대학교 컴퓨터공학과 석사학위논문, 2001

[17] 전 신우, 정 용진, "128 비트 SEED 암호 알고리즘의 고속처리를 위한 하드웨어 구현," 통신정보보호학회지, Vol. 11, No. 1, pp. 13-23, Feb. 2001

[18] L. Keliher, H. Meijer, and S. Tavares, "New Method for Upper Bounding the Maximum Average Linear Hull Probability for SPNs," Advances in Cryptology -EIROCRYPT 2001, LNCS 2045, Springer -Verlag, pp. 420-436, 2001

[19] K. Nyberg, "Differentially uniform mappings for cryptography," Advances in Cryptology, Proceedings of Eurocrypt '93, LNCS 765, T. Helleseth, ED., Springer-Verlag, pp. 55-64, 1994

[20] Serge Mister and Carlisle Adams, "Practical S-box Design," Workshop record of the workshop on selected area in Cryptography(SAC'96), Queen's University, pp. 61-76, Aug. 1996

[21] T. Jakobsen and L.R. Knudsen, "The interpolation attack on block cipher," Fast Software Encryption, LNCS 1267, E. Biham, Ed., Springer-Verlag, pp. 28-40, 1997

[22] Webster, A. and S. Tavares, "On the Design of S-Boxes," Advances on Cryptology, CRYPTO'85, pp. 523-534, 1985

[23] A.M. Youssef, Z.G. Chen and S.E. Tavares, "Construction of Highly Nonlinear Injective S-boxes With Application to CAST-like Encryption Algorithms," Proceedings of the Canadian Conference on Electrical and Computer Engineering(CCECE'97), 1997

[24] Jennifer Seberry, Xian-Mo Zhang and Yuliang Zheng, " Systematic Generation of Cryptographically Robust S-boxes," The preceedings of the First ACM Conference on Computer and Communications Security, pp. 172-182, Nov. 1993

[25] Nyberg, K., "Perfect nonlinear S-boxes," In Advances in Cryptology, EUROCRYPT'91, Vol. 547, Lecture Notes in Computer Science, Springer-Verlag, pp. 378-386, 1991

[26] J. Daemen, R. Govaerts and J. Vandewalle, "Correlation Matrixes," Fast Software Encryption, LNCS 1008, Spring-Verlag, pp. 275-285, 1994

[27] J. S. Kang, C. S. Park, S. J. Lee and J. L. Lim, "On the optimal diffusion layer with practical security against Differential and Linear Cryptanalysis," Proceedings of ICISC'99, LNCS 1787, Spring-Verlag, pp. 33-52, 1999

저자 소개

박창수(Chang-Soo Park)

1995년 인제대학교 전자공학과 졸업(공학사)

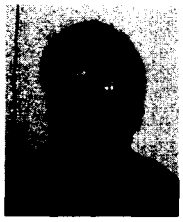
2001년 부경대학교 산업대학원 컴퓨터공학과 졸업(공학석사)

2002년-현재 부경대학교대학원 컴퓨터공학과 박사과정

※ 관심분야 : 반도체회로설계, 암호 알고리즘, 컴퓨터 구조

송홍복(Hong-Bok Song)

1983년 광운대학교 전자통신공학과 졸업
1985년 인하대학교 대학원 전자공학과 졸업(공학석사)
1985 - 1990년 : 동의공업대 전자통신과 조교수
1989 - 1990년 : 일본 구주공대 정보공학부 객원연구원
1990년 동아대학교 대학원 전자공학과 졸업(공학박사)
1994-1995년 일본 미야쟈키 대학교 전기.전자공학부 (POST-DOC)
1991년-현재 동의대학교 전자.정보통신공학부 교수
※ 관심분야 : 다치논리 이론 및 시스템 설계, VLSI 설계, 마이크로프로세서 응용



조경연(Gyeong-Yeon Cho)

1990 인하대학교 공과대학 전자공학과 정보공학전공 (공학박사)
1983-1991 삼보컴퓨터 기술연구소 책임연구원
1991-현재 부경대학교 공과대학 전자컴퓨터정보통신공학부 교수
1991-2001 삼보컴퓨터 기술연구소 비상임기술고문
1998-현재 에이디칩스 사외이사 겸 비상임기술고문
※ 관심분야 : 전산기구조, 반도체회로설계, 암호 알고리즘