8 비트 마이크로프로세서에 적합한 블록암호 알고리즘

°김 용덕*, 박 난경 **, 이 필중*
포항공과대학교 전자전기공학과*, 정보통신연구소**

A New Block Cipher for 8-bit Microprocessor

°Yong Duk Kim*, Nan Kyung Park**, Pil Joong Lee*
Dept. of Electronics and Electrical Engineering*, PIRL**, POSTECH
°kyds@oberon.postech.ac.kr*, nkpark@oberon.postech.ac.kr**, pjl@postech.ac.kr*

요 약

계산능력이 제한된 8 비트 마이크로프로세서에 적합하도록 모든 기본 연산을 8 비트 단위로 처리하는, 블록 크기는 64 비트, 키 크기는 128 비트인, Feistel 구조의 블록 암호 알고리즘을 제시한다. 이 알고리즘의 안전도는 잘 알려진 two-key triple-DES[ANSI86]나 IDEA[Lai92]와 비견할 만하며, 처리속도는 single-DES[NBS77]보다도 10~20 배 빠르다. 본 논문에서는 이 알고리즘의 설계원칙 및 안전성 분석에 대하여 설명하였고, 다른 알고리즘과의 통계적 특성 및 성능에 대해서도 비교하였다.

1. 서 론

반도체 기술의 향상으로 마이크로프로세서의 가격이 전반적으로 하락하면서 32 비트 마이크로프로세서가 주종을 이루게 되었다. 그 결과로 대부분의 암호 알고리즘도 32 비트 연산을 기본으로 하여 개발되고 있다[Riv94, Sch93]. 그러나 전자상거래를 실현할 수 있는 도구인 Smart card 는 경제성 등의 이유로 8 비트 마이크로프로세서를 기반으로 하나, 8 비트 마이크로프로세서는 기본 연산을 8 비트 단위로 처리하고 계산능력도 제한되어 있어 8 비트 전용 알고리즘이 요구된다.

이에 본 논문에서는 8 비트 마이크로프로세서에 적합한 블록 암호 알고리즘을 제안한다. 제안하는 알고리즘은 블록 크기가 64 비트이고 키 크기가 128 비트의 Feistel 구조로 기본 연산을 8 비트 단위로 처리하고 복잡한 계산을 요구하지 않는다. 또한 키와 데이터에 의존하는 연산과 inversion S-box 를 사용하여 avalanche 효과 및 DC(Differential Cryptanalysis)[BS92] 및 LC(Linear Cryptanalysis)[Mat93]에 대한 저항성을 갖도록 하였고 key scheduling 에서는 one-wayness 와 라운드 키간의 상호의존성을 만족하도록 하였다.

이에 2 장에서는 설명에 사용된 기호 및 용어를 정리하고 3 장에서는 구체적인 알고리즘의 구조 및 설계 원칙을 설명하고 4 장에서는 알고리즘의 DC 및 LC 에 대한 안전성을 살펴보고 5 장에서는 제안한 알고리즘 및 잘 알려진 다른 알고리즘 DES[NBS77], SHARK[RDP96], SQURE[DKR97]를 8 비트, 32 비트, 64

비트 마이크로프로세서에서 구현하여 성능 및 통계적 특성에 대한 분석자료를 제시한 후 6 장에서 결론을 맺는다.

2. 기호 및 용어 정리

알고리즘 연산의 기본 단위는 8 비트이고, 설명을 위해 사용되는 기호 및 용어는 다음과 같다.

$P(=(P_L, P_R))$: 평문 64 비트(P_L, P_R 은 평문의 좌우 32 비트)
$C(=(C_L, C_R))$: 암호문 64 비트(C_L, C_R 은 암호문의 좌우 32 비트)
$X^{(r)}(=(X_1^{(r)}, X_2^{(r)}, X_3^{(r)}, X_4^{(r)}))$: r 번째 라운드 함수 F 의 입력 32 비트
($X_i^{(r)}$ 은 i = 1, 2, 3, 4 에 대하여 각기 8 비트)
$Y^{(r)}(=(Y_1^{(r)}, Y_2^{(r)}, Y_3^{(r)}, Y_4^{(r)}))$: r 번째 라운드 함수 F 의 출력 32 비트
($Y_i^{(r)}$ 은 i = 1, 2, 3, 4 에 대하여 각기 8 비트)
$K^{(r)}(=(K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}, K_5^{(r)}))$: r 번째 라운드 서브키 40 비트
($K_i^{(r)}$ 은 i = 1, 2, 3, 4, 5 에 대하여 각기 8 비트)

$\odot$: multiplication modulo 2^8+1
$\oplus$: bitwise XOR
$\boxplus$: addition modulo 2^8
S : S-box, $S(x) = x^{-1}$ in $GF(2^8)$의 affine transform[DKR97]
$<<_c n$: circular left shift by n 비트

n 비트 블록이 있을 때 little endian 에서의 최상위비트(MSB : Most Significant Bit)를 0 번 비트로 나타내고 최하위비트(LSB : Least Significant Bit)를 n-1 번 비트로 나타낸다.

3. 알고리즘의 구조 및 설계 원칙

이 장에서는 제안하는 알고리즘의 구조 및 설계 원칙을 설명한다.

3.1 Feistel 구조

새로 제안하는 블록암호 알고리즘의 블록크기는 64 비트, 키 크기는 128 비트이고 그림 3.1 과 같은 16 라운드 Feistel 구조 알고리즘이다. r 번째 라운드에서 블록을 좌우 32 비트로 나눈 후 라운드 함수 F 는 32 비트 $X^{(r)}$과 서브키 40 비트 $K^{(r)}$를 입력으로 하여 32 비트 $Y^{(r)}$을 출력한다.

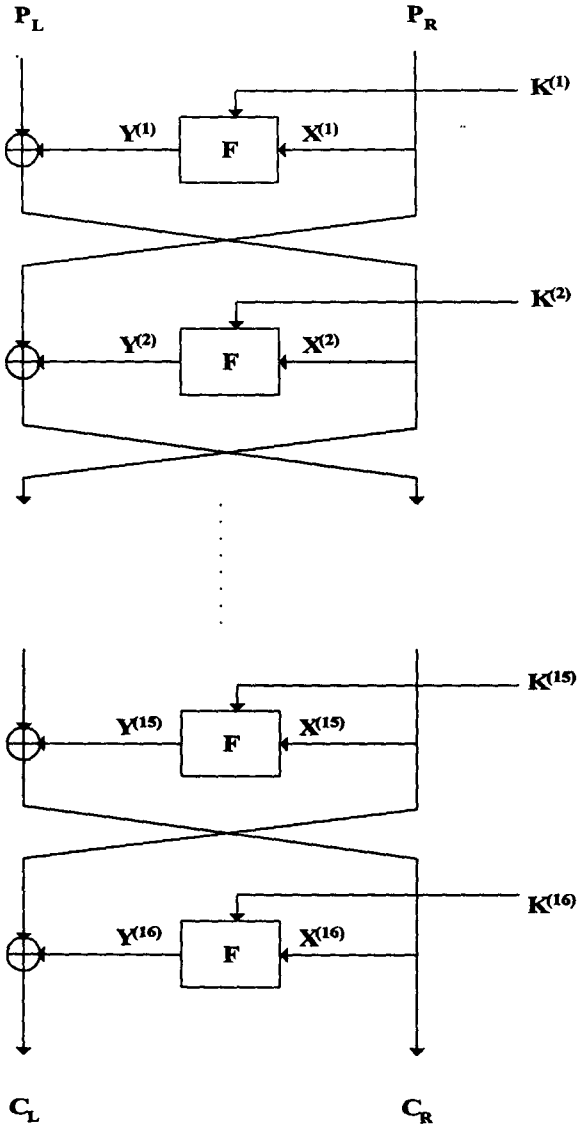

그림 3.1 Feistel 알고리즘 구조

3.2 라운드 함수 F

라운드 함수 F의 구조는 그림 3.2와 같다. $X^{(r)}$과 $K^{(r)}$은 각각 8비트씩 $X_1^{(r)}, X_2^{(r)}, X_3^{(r)}, X_4^{(r)}$과 $K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}, K_5^{(r)}$로 나뉘어져 연산들이 이루어진다.

F 라운드 함수 구성시의 설계원칙은 다음과 같다.

- 8비트 연산

Smart card 와 같은 8 비트 마이크로프로세서에 적합하도록 하기 위해 기본 연산의 단위를 8 비트로 하였다.

- Confusion /Diffusion
  키를 모르고 평문과 암호문간의 연관성을 찾기 힘들고 평문 및 키의 변화가 암호문에 골고루 영향을 주도록 하여 통계적 특성에 의한 공격을 힘들게 하였다[Lai92, Sha49].
- DC 및 LC 에 대한 저항성
  현재까지 알려진 가장 강력한 cryptanalysis 인 DC 및 LC 에 대한 저항성을 갖도록 하였다.

3.2.1 Key Addition

입력 $X^{(r)}$과 네 개의 서브키($K_1^{(r)}, K_2^{(r)}, K_3^{(r)}, K_4^{(r)}$)을 bit wise XOR 로 key addition 하여 $A_1^{(r)}, A_2^{(r)}, A_3^{(r)}, A_4^{(r)}$를 얻는다.

3.2.2 데이터, 키 의존 연산

Key addition 의 결과인 $A_1^{(r)}, A_2^{(r)}, A_3^{(r)}, A_4^{(r)}$과 $K_5^{(r)}$을 이용하여 새로운 값 $U^{(r)}$을 다음 계산으로 구한다.

$$U^{(r)} = K_5^{(r)} \odot (A_1^{(r)} \boxplus A_2^{(r)} \boxplus A_3^{(r)} \boxplus A_4^{(r)})$$

$U^{(r)}$은 key addition 된 데이터와 $K_5^{(r)}$에 대한 정보를 포함하므로 $U^{(r)}$을 계산하는 연산을 데이터, 키 의존 연산이라고 하기로 한다. 이 연산에서 $A_1^{(r)}, A_2^{(r)}, A_3^{(r)}, A_4^{(r)}$를 addition modulo 2^8 하므로 입력의 변화가 모든 출력에 영향을 주는 diffusion 및 avalanche 효과가 생긴다[HT95, HT96]. 또한 $K_5^{(r)}$를 XOR 에 대해 비선형적인 연산인 multiplication modulo 2^8+1 하여 키의 정보를 포함하게 하므로 DC 및 LC 을 이용한 공격을 어렵게 하였다.

3.2.3 S-box

S-box 는 substitution box 로 DC 및 LC 에 대해 저항성을 가지도록 하였다. S-box 는 inversion mapping x^{-1} over $GF(2^8)$이나 고정된 점들(0, 1)을 제거하기 위하여 affine transformation 하였다[DKR97]. 이 S-box 는 DC, LC, interpolation attack[JK97]에 대해 안전성을 제공하는 좋은 특성이 있어 기존의 알고리즘인 SHARK, SQUARE 등에서 쓰였다.

3.2.4 Permutation

Permutation 은 평문 및 키의 변화가 암호문에 골고루 영향을 주도록 하였다. 입출력 32 비트 각각을 2 비트씩 16 개로 생각하여 permutation 을 표 3.1 과 같이 설계하였다.

출력	0	1	2	3	4	5	6	7
입력	3	6	10	13	0	7	11	14
출력	8	9	10	11	12	13	14	15
입력	1	4	8	15	2	5	9	12

표 3.1 permutation

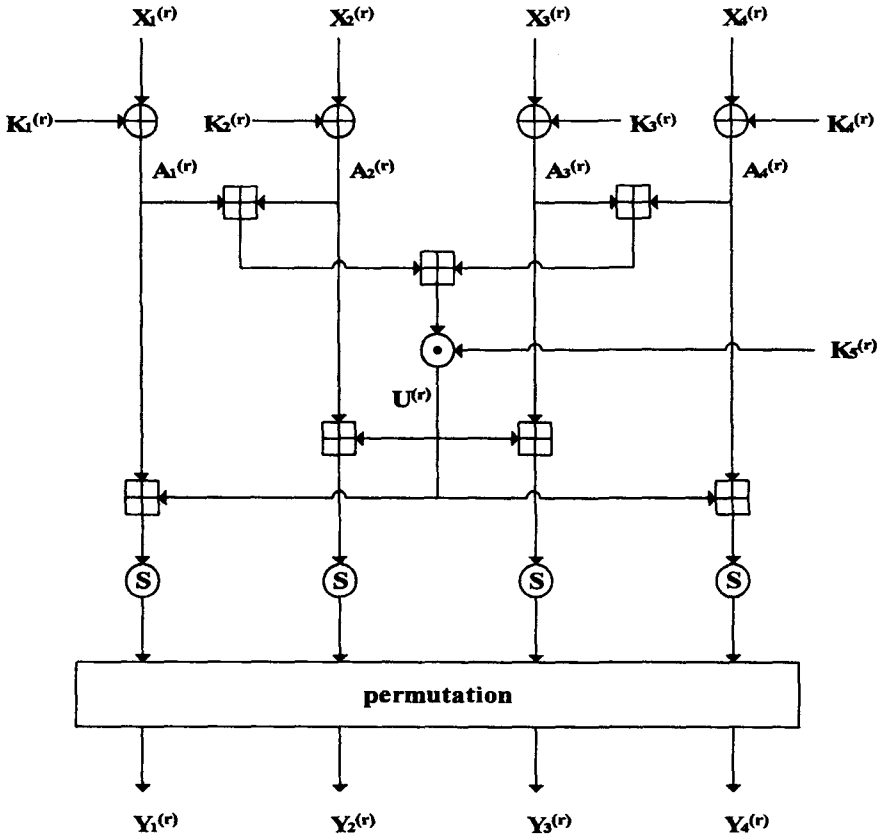

그림 3.2 r 번째 라운드 함수 F

3.3 Key Scheduling

Key scheduling은 그림 3.3과 같이 128비트 키 K를 입력으로 하여 40비트 서브키 16개를 생성한다. 이 때 설계원칙은 다음과 같다.

- Key size
  128비트로 하여 DES에 적용한 것과 같은 exhaustive key search attack[Wie93]을 어렵게 하였다.
- One-way ness
  One-wayness를 만족하면 몇 개의 라운드 서브키만을 알고 키나 다른 라운드 서브키를 알 수 없다. 이러한 특성을 만족하지 않으면 weak key를 찾거나 DC 및 related-key attack이 쉽다[Bih93, BS92, KSW96].
- 라운드 서브키 간의 상호 의존성
  라운드 서브키가 상호 독립적이면 related-key attack에 약하므로 상호 의존성을 갖도록 하였다[KSW96].
- 알려진 공격(related-key attack, differential attack)에 대한 저항성
  모든 키 비트가 모든 라운드에 영향을 주도록 하여 서브키의 avalanche 효과를 최대로 하고 선형 키 schedule을 피하도록 하여 알려진 공격에 대한 저항성을 갖도록 하였다[KSW96].

내부구조를 그림 3.3의 g_i, h_i($i = 1, 2, \ldots, 5$)로 나누어 살펴보면 다음과 같다. 이 때 g_i의 입력 128비트 K를 B로 하자(B=K). g_i의 입력 128비트 B에 대하여 출력은 수정된 128비트 B이다. h_i의 입력은 g_i의 출력 128비트 B이고 B에서 선택된 80비트의 각 40비트 $K^{(2i-1)}$, $K^{(2i)}$가 $i = 1$이면 출력이 되나 $i \geq 2$이면 $K^{(2i-3)}$, $K^{(2i-1)}$과 XOR되어 새로운 $K^{(2i-1)}$, $K^{(2i)}$이 출력이 된다.

- g_i를 우선 살펴보자.

  $B = K$

  입력 128비트 B를 8비트씩 16개의 $B_1, B_2, \ldots, B_{15}, B_{16}$로 나눈다.

  $$V_1 = \sum_{k=1}^{8} B_{2k-1}, \quad V_2 = \sum_{k=1}^{8} B_{2k}$$

  $V_1 = V_1 \odot V_2$

  $V_2 = V_1 \oplus V_2$

  $B_0 = B_{16}$

  $B_j = (V_1 \oplus B_{j-1} + V_2 \oplus B_j) <<_c 1 \quad \text{for } j = 1, 2, \ldots, 16$,

  출력은 수정된 128비트 B이다.

- h_i를 살펴보자.

  $K_s^{(2i-1)} = B_{((2i+3s-1) \bmod 16)} \quad \text{for } s = 1, 2, \ldots, 5$

  $K_s^{(2i)} = B_{((2i+3s) \bmod 16)} \quad \text{for } s = 1, 2, \ldots, 5$

  단 $i \geq 2$ 일 때

  $K_s^{(2i-1)} = K_s^{(2i-1)} \oplus K_s^{(2i-3)}$

  $K_s^{(2i)} = K_s^{(2i)} \oplus K_s^{(2i-2)}$

키 128 비트 입력 B 에 대하여 g_i 에서의 연산으로 새로운 B 를 얻은 후 B 의 8 비트 16 개 중 8 비트 10 개의 B_i 를 선택하였다. 또한 g_i, h_i 의 연산 결과를 계속해서 사용하고 서브키 간의 XOR 연산을 하였다. 그 결과 one-wayness 및 라운드.키간의 상호의존성 등을 만족하여 기존 공격에 대하여 저항성을 갖도록 하였다.

또한 weak key 는 암호화와 복호화가 같게 되어 기존 블록 암호 알고리즘의 문제가 되었다[DGV93, MS88]. 그러나 제안된 알고리즘에서는 DES 와 같은 구조적 특성에 의한 weak key 가 존재하지 않는다. 또한 128 비트가 모두 0 이거나 모두 1 인 경우에 점검한 결과 weak key 가 발견되지 않았다.

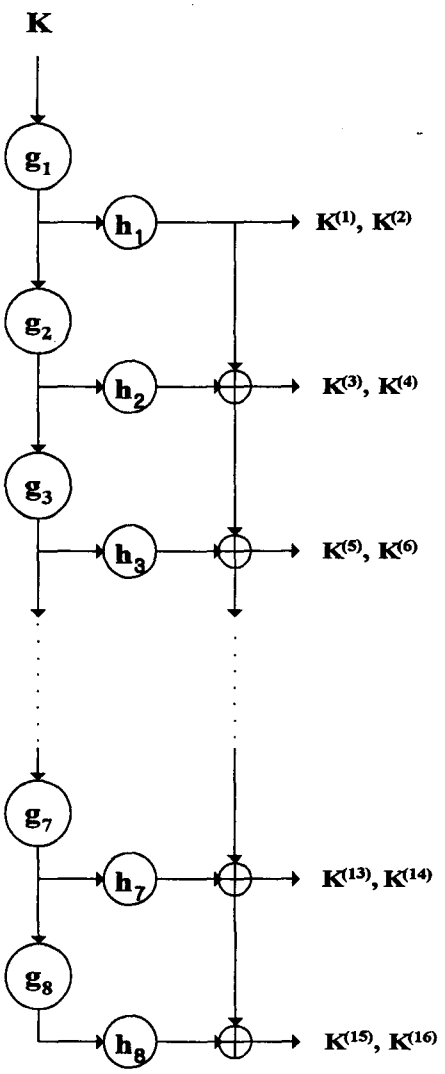

그림 3.3 Key scheduling

4. DC 및 LC에 대한 안전성

알고리즘에서 DC 및 LC에 대한 저항성을 갖도록 한 부분은 3.2.2 데이터, 키 의존 연산과 3.2.3 S-box이다. $U^{(r)}$을 만들 때 $K_5^{(r)}$를 multiplication modulo 2^8+1 하여 키의 정보가 포함되도록 하였다. 그러므로 DC에서는 고정된 키에 대한 입출력간의 차(difference)를 구해도 키를 제거하지 못하고 LC에서는 multiplication modulo 2^8+1에 대한 입출력 및 키에 대한 선형근사식을 만들기가 힘들어 DC 및 LC에 대한 저항성을 갖게 된다[BS92,Mat93,Ada97]. 이 값 $U^{(r)}$을 $A_1^{(r)}$, $A_2^{(r)}$, $A_3^{(r)}$, $A_4^{(r)}$과 다시 addition modulo 2^8 하므로 모든 8비트 블록들에 대해서 DC 및 LC을 이용한 공격이 어렵게 된다.

Inversion mapping S-box의 DDT(Difference Distribution Table)[BS92]의 최대값은 4이고 LDT(Linear Distribution Table)[Mat93]의 최대값은 144이다. 따라서 DP^s[Mat96]$\leq(1/2^6)$이고 LP^s[Mat96]$\leq(1/2^6)$이다. 그러나 제안된 알고리즘에서는 S-box 이전에 addition modulo 2^8 연산이 있으므로 addition modulo 2^8과 S-box를 하나의 연산 S'로 생각하여 DDT 및 LDT을 구하면 DC 및 LC에 대한 안전성을 측정할 수 있다. 그러나 $U^{(r)}$에는 키에 대한 정보가 포함되어 있어 DDT 및 LDT를 구하기 힘들므로 키와 무관하게 단순히 입력을 2^8개만으로 한정하였다.

DDT의 최대값은 12이고 LDT의 최대값은 184이므로 S'의 $DP^{s'}\leq(3/2^6)$이고 $LP^{s'}\leq(81/2^{10})$이 된다. 제안된 알고리즘에서는 라운드 함수 F의 입력이 8비트 블록 4개로 이루어져있고 Feistel 구조 또한 DES에서 사용되었던 기본 형태이므로 DP_F[Mat96]$\leq(3/2^6)^8$이고 LP_F[Mat96]$\leq(81/2^{10})^8$이다. 또한 r 라운드 알고리즘 전체에 대한 differential prob. DP(r)[Mat96]$\leq(1/2^{32})^{r/4}$이고 linear hull prob. LP(r)[Mat96]$\leq(9^8/2^{40})^{r/4}$이다. 따라서 SQUARE에서처럼 DC 및 LC을 이용한 공격이 exhaustive key search attack(확률 2^{-127})보다 비효율적이기 위해서는 DP(r)$\leq 2^{-127}$, LP(r)$\leq 2^{-64}$이어야 하므로 라운드 수가 최소 16라운드 이상 이어야 한다[DKR97, KSS97, Mat96].

5. 알고리즘의 분석

이 장에서는 제안하는 알고리즘을 구현하여 통계적 특성 및 8비트 마이크로프로세서의 수행속도를 기존의 다른 알고리즘인 DES, SHARK, SQUARE와 비교한다. 제안하는 알고리즘은 블록 크기가 64비트, 키 크기가 128비트이고 DES는 블록 크기가 64비트, 키 크기가 56비트이고 SHARK는 블록 크기가 64비트, 키 크기가 128비트이며 SQUARE는 블록 크기가 128비트, 키 크기가 128비트이다. DES는 현재 가장 널리 쓰이는 알고리즘이고 SHARK와 SQUARE는 기본 연산을 8비트단위로 하나 SHARK는 64비트 마이크로프로세서에서 효율적이고 SQUARE는 8비트 마이크로프로세서에서 좀 더 효율적이므로 비교대상으로 하였다[RDP96, DKR97].

5.1 통계적 특성

이 장에서는 제안된 알고리즘의 통계적 특성을 블록크기가 64 비트인 DES, SHARK 와 비교한다. 통계적 특성을 측정하기 위해 64 비트 random sequence 10,000 개를 만들어 각 알고리즘에 적용을 하였다.

알고리즘	통계적 특성			
	Frequency -1 test	Poker-8 test	Serial test	Runs test(L=15)
	% of n1 where n1 <24 or n1>40 with 5% Significance Level	% of Failed Samples with 5% Significance Level	% of Failed Samples with 5% Significance Level	% of Failed Samples with 5% Significance Level
제안된 알고리즘	3.12	10.05	6.70	8.96
DES	3.78	10.34	7.02	8.91
SHARK	3.04	8.95	6.92	8.48

표 5.1 통계적 특성 비교

표 5.1 에서 frequency-1 test, poker-8 test 는 64 비트 sequence 의 임의성(randomness)을, serial test, runs test 는 sequence 의 독립성을 알아보는 것이다[최봉대 92]. 즉 frequency-1 test 에서는 sequence 가 random 하면 0 의 개수(n0)와 1 의 개수(n1) 모두 32 이어야 하므로 normal 분포 N(32, 8)에 대하여 유의 수준(significant level) 5%의 양측 검정하였다. 또한 poker-8 test, serial test, runs test 는 각각 chi-square 분포 χ^2(자유도 8), χ^2-분포(자유도 2), χ^2-분포(자유도 30)에 대하여 유의 수준 5%의 검정을 하였다[최봉대 92, BJ77]. 표 5.1 의 결과는 위 네 가지 test 들을 10,000 개의 random sequence 에 대하여 적용했을 때 기각역(critical region value)에 속하는 sequence 들의 개수를 백분율로 나타낸 것이므로 수치가 작을수록 알고리즘이 임의성 및 독립성을 갖는다고 볼 수 있다. 결론적으로 제안된 알고리즘이 DES 에 비해서는 runs test 을 제외한 frequency-1 test, serial test, poker-8 test 에서 다소 우수하나 SHARK 에 비해서는 serial test 을 제외한 frequency-1 test, poker-8 test, runs test 에서 다소 좋지 않음을 알 수 있다[GDC96, GDNC96].

5.2 수행속도

8 비트 마이크로프로세서인 8051(Archimedes C compiler v4.1b/DOS 사용), 32 비트 마이크로프로세서인 Pentium 200MHz(Visual C++ 사용)와 64 비트 마이크로프로세서인 Sun SPARC 20 의 100MHz(GNU C compiler v2.7 사용)에서 수행하였으며 입력 화일의 크기는 1Mbytes 이고 100 개의 다양한 키 값에 대하여 수행속도의 평균값을 측정하였다. DES 는 [SP89]의 Appendix B1 의 코드를 수정하여 구현했으며 SHARK 는 Vincent Rijmen(1995 년 12 월 작성, Copyright (C): KULeuven)의 코드를 이용하였고 SQUARE 는 Joan Daemen, Vincent Rijmen(1997 년 2 월 7 일 작성)의 코드를 이용하였다. SHARK, SQUARE 의 코드는 ftp://ftp.esat.kuleuven.ac.be/pub/COSIC/rijmen 에서 구할 수 있다.

알고리즘	수행속도(Kbits/sec)[DES 에 대한 상대 속도]		
	8051	Pentium 200MHz	SPARC 20 100MHz
	Archimedes C v4.1b/DOS	Visual C++ 5.0	GNU C v2.7
제안된 알고리즘	0.860[17.92]	3631[10.32]	785[6.76]
DES	0.048[1]	352[1]	116[1]
SHARK	*	450.7[1.28]	354[3.05]
SQUARE	0.366[7.63]	1005[2.86]	272[2.34]

표 5.2 수행속도 비교

표 5.2 의 수행속도를 비교해 보면 제안하는 알고리즘이 다른 알고리즘보다 수행속도의 빠른 정도가 64 비트, 32 비트 마이크로프로세서보다 8 비트 마이크로프로세서에서 높음을 알 수 있다. 또한 SHARK 는 64 비트 마이크로프로세서에서 효율적이므로 Pentium 보다 SPARC 20 에서 빠른 정도가 개선되었음을 알 수 있다. 반면 SQUARE 는 8 비트 마이크로프로세서에서 효율적이므로 Pentium, SPARC 20 보다 8 비트 마이크로프로세서인 8051 에서 빠른 정도가 개선되었다. 결론적으로 제안된 알고리즘이 다른 알고리즘보다 빠를 뿐만 아니라 8 비트 마이크로프로세서에 더 적합함을 알 수 있다.

6. 결론

제안하는 블록 암호 알고리즘과 기존의 알고리즘 DES, SHARK, SQUARE 를 구현하여 8 비트, 32 비트, 64 비트 마이크로프로세서에서 수행속도를 비교한 결과 성능이 우수하였을 뿐만 아니라 8 비트 마이크로프로세서에 적합함을 알 수 있었다. 또한 기존 공격 방법에 대해 안전하고 통계적 특성도 기존 알고리즘과 비슷함을 알 수 있었다. 차후에 수행속도를 좀 더 개선하기 위해 구현방법의 최적화 및 안전도 분석의 개선이 필요하다.

* SHARK 의 소스코드는 64 비트 기반으로 구현되어 있어 8051 마이크로 프로세서에서 구현할 수 없었다.

참고 문헌

[최봉대92] 최 봉대, 신 양우, 한 동환, 최 두일, "Randomness 특성 분석에 관한 연구," *한국전자통신연구소 데이터 보호의 기반 기술 연구 보고서*, 한국 과학 기술원, 12/92.

[Ada97] Carlisle M. Adams, "Constructing Symmetric Ciphers Using the CAST Design Procedure," *Designs, Codes and Cryptography*, vol. 12, no. 3, pp.71-104, November 1997.

[ANSI86] ANSI X9.9 (Revised), "American National Standard for Financial Institution Message Authentication (Wholesale)," American Bankers Association, 1986.

[BJ77] Gouri. Bhattacharyya, Richard A. Johnson, "*Statistical Concepts and Methods*," John Wiely & Sons, Inc. 1977.

[BS92] Eli Biham, Adi Shamir, "Differential Cryptanalysis of the full 16-round DES," *Advances in cryptology - CRYPTO'92*, LNCS 746, Springer-Verlag, pp.487-496, 1993.

[Bih93] Eli Biham, "New Types of Cryptanalytic Attacks Using Related Keys," *Advances in cryptology - EUROCRYPT '93*, Springer-Verlag, pp. 398-409, 1994.

[DGV93] J. Daemen, R. Govaerts and J. Vanderwalle, "Weak Keys for IDEA," *Advances in cryptology -Eurocrypt93*, Springer-Verlag, 1993.

[DKR97] Joan Daemen, Lars Knudsen, Vincent Rijmen, "The Block Cipher SQUARE," *PreProc. of 4th Fast Software Encryption Workshop*, Springer-Verlag, pp.137-151, 1997.

[GDC96] Helen Gustafson, Ed Dawson, Bill Caelli, "Comparison of Block Ciphers," private communication, 1996.

[GDNC96] H.Gustafson, E.Dawson, L.Nielsen, B.Caelli, "Computer Package for Measuring the Strength of Encryption Algorithms," private communication, 1996.

[HT95] H M. Heys, S. E. Tavares, "Avalanche Characteristics of Substitution-Permutation Encryption Networks," *IEEE Trans. Computers*, Vol. 44, pp. 1131-1139, September 1995.

[HT96] H M. Heys, S. E. Tavares, "Substitution-Permutation Networks Resistant to Differential and Linear Cryptanalysis," *J. of Cryptology '96*, Vol. 9, pp. 1-19, 1996.

[JK97] T. Jakobsen, Lars R. Knudsen, "The Interpolation Attack on Block Ciphers," *PreProc. of 4th Fast Software Encryption Workshop*, LNCS, Springer-Verlag, pp. 28 - 40, 1997.

[KSS97] Y. Kaneko, F. Sano, K. Sakurai, "On Provable Security against Differential and Linear Cryptanalysis in Generalized Feistel Ciphers with Multiple Random Functions," *Proc. of SAC '97 - Workshop on Selected Areas in Cryptography*,1997.

[KSW96] John Kelsey, Bruce Schneier, David Wagner, "Key-Schedule Cryptanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES," *Advances in Cryptology-CRYPTO '96*, Springer-Verlag, 1996.

[Lai92] Xuejia Lai, "*On the Design and Security of Block Ciphers*," Hartung-Gorre Varlag Konstanz, 1992.

[MS86] J.H.Moore and G.J. Simmons, "Cycle Structure of the DES with Weak and SemiWeak Keys," *Advances in Cryptology-Crypto'86*, 1986.

[Mat93] M. Matsui, "Linear Cryptanalysis Method for DES Cipher," *Advances in Cryptology - EUROCRYPT '93*, Springer-Verlag, vol. 765, pp.386 - 397, 1994.

[Mat94] Mitsuru Matsui, "On Correlation Between the Order of S-boxes and the strength of DES," *Advances in Cryptology - EuroCrypt'94*, LNCS 950, Springer-Verlag, pp. 366-375, 1994.

[Mat96] Mitsuru Matsui, "New Structures of Block cipher with provable security against differential and linear cryptanalysis," *Proc. of 3rd Fast Software Encryption Workshop*, Cambridge, U.K, LNCS 1039, Springer-Verlag, Berlin, pp.205 - 218, 1996.

[NBS77] National Bureau of Standards, NBS FIPS PUB 46, "Data Encryption Standard," U. S. Department of Commerce, Jan. 1997.

[RDP96] Vincent Rijment, Joan Daemen, Bart Preneel, Antoon Bosselaers, "The Cipher SHARK," *Proc. of 3rd Fast Software Encryption Workshop*, Springer-Verlag, pp.99-111, 1996.

[Riv94] R.L.Rivest, "The RC5 Encryption Algorithm," *Proc. of 2nd Fast Software Encryption Workshop*, Springer-Verlag, pp.86-96, 1994.

[SP89] J. Seberry, J. Pieprzyk, "*Cryptography : An Introduction to Computer Security*," Prentice Hall, 1989.

[Sch93] B. Schneier, "Description of a New Variable-Length Key, 64-Bit Block Cipher(Blowfish)," *Proc. of 1st Fast Software Encryption Workshop*, Springer-Verlag, pp.191-204, 1993.

[Sha49] C. E. Shannon, "Communication Theory of Secrecy Systems," *Bell System Technical Journal*, 28:657-715, 1949.

[Wie93] M.J. Wiener, "Efficient DES Key Search," *Advances in Cryptology - Crypto'93* rump sessions, August 20, 1993.