PKC 블록 암호 알고리즘

김 길 호*, 조 경 연**

*부경대학교 대학원 컴퓨터공학과

PKC Block Cipher Algorithm

Gil-Ho Kim*, Gyeong-Yeon Cho**

Dept. of Computer Engineering, Graduate School, Pukyong National Univ.

E-mail : vnlqpcdd@daum.net

요 약

선진국들은 공모 사업을 통해 많은 블록 암호가 개발되었으나 국내에서 개발된 블록 암호들은 크게 주목 받지 못했다. 블록 암호 개발은 기존의 암호와 차별성, 안전성 그리고 여러 플랫폼에서의 효율성이 중시되는데 이러한 조건을 다 만족하는 것은 쉽지 않기 때문이다. 본 논문은 128bit 블록 단위에서 128, 196, 256bit 키를 사용하는 새로운 블록 암호 알고리즘을 제안한다. 기존의 블록 암호 알고리즘은 SPN(Substitution-Permutation Network)구조, Feistel Network구조 등인데 본 논문에서 제안한 블록 암호 알고리즘은 변형된 Feistel Network구조로 입력 값 전체에서 선택된 32bit 만 update 된다. 이러한 구조적 특성은 기존은 블록 암호 알고리즘들과 큰 차별이 되고 있다. PKC블록 암호 알고리즘은 국제 표준 블록 암호 알고리즘인 AES와 국내 표준 블록 암호 알고리즘인 SEED와 수행 속도 면에서 동등하거나 많이 개선된 것을 보이고 있다. 이러한 특성을 이용하면 제한된 환경에서 수행해야 하는 스마트카드 와 같은 분야에 많이 활용 될 수 있을 것이다.

Key word : Feistel Network, SPN, 소수 다항식(irreducible polynomial), S-box

Ⅰ. 서 론

지식기반 정보 사회로 빠르게 진행함에 따라 정보보호에 대한 인식이 점차 높아지고 있으며, 정보보호를 위한 기술적 대응조치로 암호기술이 일반적이다. 암호의 사용이 활성화됨에 따라 선진국들은 자국의 기술을 대외에 과시하고 정보보호 제품의 수출입을 규제하는 등의 목적으로 자국의 표준 암호를 갖고자 많은 노력을 기울이고 있다. 미국의 경우 NIST[1](National Institute of Standards and Technology)에서 새로운 블록 암호 알고리즘 AES(Advanced Encryption Standard)[2]에 대한 공모 사업을 주관해서 Rijndeal을 선정 했으며, 유럽 연합(EU)에서도 회원국들을 위한 암호 알고리즘 공모사업인 NESSIE[3](New European Schemes for Signatures. Integrity. and Encryption)를 진행하여 암호 알고리즘의 표준화를 추진하고 있다. 한편 일본도 2003년부터 CRYPTREC[4](Cryptography Research & Evaluation Committee)를 진행 중에 있다.

우리나라도 1999년 9월 자체 개발한 SEED[5]를 국가 표준으로 제정하였다.

본 논문에서는 블록 암호 알고리즘 PKC(Pu Kyong Cipher)를 제안한다. PKC 암호 알고리즘은 기존의 Feistel Network구조와 다른 입력 128bit에 대해서 96bit를 가지고 32bit만 값을 바꾸는 과정을 반복 수행한다. 이러한 구조는 다른 블록 암호 알고리즘과 큰 차별을 보이고 있다. 키는 128, 192, 256bit 3종류를 사용 할 수 있도록 구성되었다. 본 논문에서는 128bit 키를 가지고 설명한다. 암호/복호는 전체 2라운드로 구성 되었으며, 암호과정에서 8bit S-box 한 개 와 XOR 연산만으로 구성되었으며, 라운드마다 16개의 32bit 라운드 키를 사용한다.

본 논문에서 제안한 PKC 암호 알고리즘은 2장에서 알고리즘의 전체 구조, 각각의 라운드 함수에 대해서 자세히 소개하고, 라운드 키 생성과정 또한 설명한다. 3장에서 암호/복호화 과정을 자세히 설명하며 4장에서 결론을 기술한다.

II. 알고리즘구조

1. PKC 블록 암호 알고리즘의 특징

- 변형된 Feistel Network 구조
- 128bit 입/출력
- 128/192/256bit 키
- 2 라운드

본 논문에서 사용된 기호

⊕ : XOR 연산
⊞ : mod 32 덧셈 연산
⓵ : OR 연산
⊙ : AND 연산
ㅣ : NOT 연산
⋘n : n-bit 오른쪽회전 연산

2. 라운드 함수

각 라운드는 다음과 같이 구성 되어 있다.

- 라운드 키 덧셈(Add round key layer): 각 라운드에서 입력 128bit 와 라운드 키 128bit 간의 XOR 연산 수행.
- 초기화(Whitening layer): 입력 96bit 와 라운드 키 32bit 간의 Hash 함수를 통한 초기화 과정으로 구성.
- 확산 계층(Diffusion layer): S-box를 통과한 8bit단위의 입력 4개를 각각 mod 32 덧셈과 라운드 키와 XOR연산으로 확산 함수 구성.
- 치환 계층(Substitution layer): 8bit S-box 1개로 구성.

2.1 초기화 계층

초기화 계층은 라운드 키 덧셈수행 후 생성된 128bit 중 32bit 오른쪽 회전연산으로 선택된 96bit와 라운드 키 32bit간의 XOR, AND, OR연산으로 32bit를 생성한다. 이와 같은 과정을 4회 반복으로 최종 128bit가 생성된다. 그림 1은 초기화 과정을 설명하고 있다.

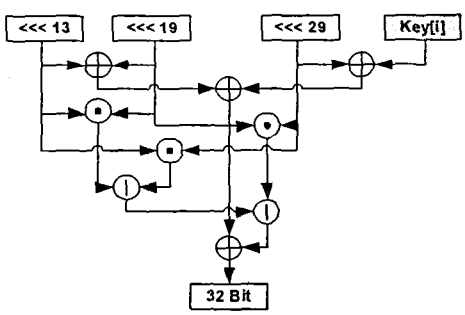

그림 1 Hash 함수

2.2 확산 계층

확산 계층 설계에 있어서 중요한 고려사항으로 안전성과 여러 환경에서의 효율적인 구현 가능성이다. PKC 암호 알고리즘은 다음과 같은 설계 방향을 가지고 설계되었다.

- AES[6]에서 제시된 강력한 분석 방법들에 대하여 충분한 내성을 가져야 한다.
- 8bit 32bit 소프트웨어 및 하드웨어 구현에 적합해야 한다.
- 동종의 확산 함수 중에서 안전성과 효율성을 고려할 때 우수해야 한다.

그림 3의 F 함수에서 첫 번째 S-box를 통과한 4개의 8bit는 각각 mod 32 덧셈을 수행한다. 그리고 각각의 32bit는 XOR연산을 수행하고 마지막으로 라운드 키 32bit와 다시 XOR연산을 수행한다. 확산 계층의 수행 후 SCONST(0x6beccd2f)의 추가 이유는 S-box의 weak point를 없애기 위함이다. S-box의 weak point는 다음과 같다.

- S-box(a) → a
- S-box(a) → -a
- S-box(a) → ~a

2.3 치환 계층

치환 계층은 8bit 입/출력 S-box로 구성된다. S-box는 다음의 성질을 만족하도록 선택되었다.

입출력 크기: 8Bit
최대 차분/선형 확률: 2^{-6}
대수적 차수: 7
고정점 반고정점이 없을 것

이러한 성질을 만족하는 것들로는 유한체 $GF(2^8)$상의 함수 x^{-1}에 아핀 변환을 취한 형태가 널리 사용되고 있다.[7] x^{-1} 와 특성이 같은 것으로 x^{-2n} 인데 n = 0,...,7이 있다.

본 논문에서 사용한 S-box는 유한체 $GF(2^8)$상의 함수 x^{-1}에 아핀 변환을 사용 했는데 이때 소수 다항식(irreducible polynomial)을 가변적으로 사용 했다.

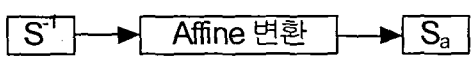

그림 2 S-box 생성원리

그림 2는 S-box생성 원리를 나타낸 것으로 첫 번째 통과 S-box에서는 소수 다항식 $P(x) = x^8+x^4+x^3+x^2+1$이고, 아핀 변환 후, 생성된 S_a의 소수 다항식 $P(x) = x^8+1$이다.

$$S_a = S^{-1} \cdot 0x1f \quad (1)$$

식(1)은 그림 3 F 함수에서 첫 번째 통과 S-box의 생성 원리이다. 두 번째 통과 S-box는 첫 번째와 다르게 $P(x) = x^8+x^7+x^5+x+1$이고, 아핀 변환 후, 생성된 S_a의 소수 다항식 $P(x) = x^8+x^7+x^6+x^2+1$이다.

$$S_a = S^{-1} \cdot 0x38 \oplus 0x94 \quad (2)$$

식(2)는 두 번째 통과 S-box의 생성 원리이다. 확산 계층과 치환 계층의 수행은 본 논문에서 제안한 알고리즘의 핵심으로 그림 3은 F 함수를 통해 확산 계층과 치환 계층 구현한 것으로 선택된 입력 96bit와 첫 번째 라운드 키 32bit가 논리연산을 통한 초기화 과정을 거친 후, 치환 → 확산 → 치환의 과정을 보여주고 있다.

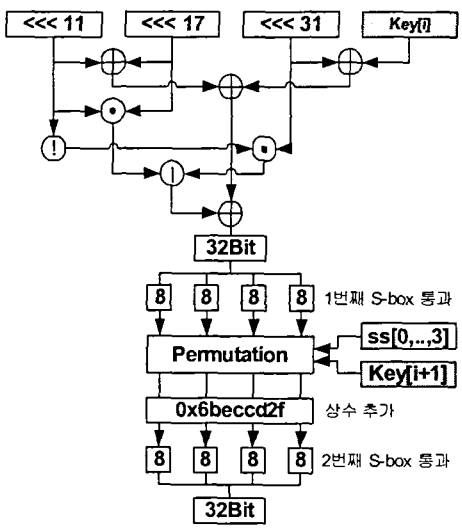

그림 3 F 함 수

2.4 라운드 키 생성

라운드 키 생성 알고리즘은 다음과 같은 설계 기준으로 설계 되었다.

- 연속된 두 라운드에서 사용된 값은 적어도 하나는 달라야 한다.
- 같은 두 값을 사용하여 생성한 두 라운드 키의 일부분으로 두 값의 일부를 예측 할 수 없어야 한다.
- 암호/복호 시 라운드 키 생성에 소요되는 시간이 적어야 한다.

라운드 키 생성은 초기화 과정을 통해 4개의 32bit 키 rkey[0], rkey[1], rkey[2], rkey[3]을 생성한다. PKC 암호 알고리즘에서는 2라운드를 제안하고, 라운드 마다 16개의 라운드 키가 필요하므로 전체 암/복호화 과정에서 총 32개의 32bit 라운드 키가 필요하다.

초기화 과정에서 생성된 4의 키 중 rkey[3]을 이용해서 순차적으로 전체 라운드 키를 생성 한다. 암/복호화 라운드 키는 다음과 같다.

```
rkey[4] ^= S-box(rkey[3])
rkey[5] = rkey[4] ^ (S-box(rkey[3]) ≪ 11)
rkey[6] = rkey[5] ^ (S-box(rkey[3]) ≪ 17)
rkey[7] = rkey[6] ^ (S-box(rkey[3]) ≪ 29)
rkey[8] ^= S-box(rkey[7])
rkey[9] = rkey[8] ^ (S-box(rkey[7]) ≪ 11)
rkey[10] = rkey[9] ^ (S-box(rkey[7]) ≪ 17)
rkey[11] = rkey[10] ^ (S-box(rkey[7]) ≪ 29)
                    .
                    .
                    .
rkey[28] ^= S-box(rkey[27])
rkey[29] = rkey[28] ^ (S-box(rkey[27]) ≪ 11)
rkey[30] = rkey[29] ^ (S-box(rkey[27]) ≪ 17)
rkey[31] = rkey[30] ^ (S-box(rkey[27]) ≪ 29)
```

여기서 S-box()는 32bit 값을 8bit단위로 나누어 S-box를 통한 치환을 수행하는 연산이다.

III. 암호 / 복호 과정

PKC 암호 알고리즘은 ASIC등의 하드웨어 구현에서 효율성을 위하여 작은 게이트수를 사용하도록 설계 했으며, 스마트카드 등의 소프트웨어 구현에 적합 하도록 작은 메모리 요구량을 가지도록 설계하여, 다양한 플랫폼에 적용 가능 하도록 설계 되었다.

그림 4는 PKC암호 알고리즘의 1라운드 암호화 과정을 설명하는 것으로 입력 128bit 에 라운드 키 128bit 가 XOR 연산을 수행 후 2.1절에서 설명한 초기화 과정을 수행하는 Hash 함수를 통과하게 된다. Hash 함수의 입력으로 선택된 32bit 입력 3개와 라운드 키 32bit로 총 128bit 가 되고, 출력은 32bit로 출력 결과는 입력에서 선택 되지 않은 32bit와 XOR연산을 수행하여 32bit만이 update 된다. 이러한 과정을 총 4회 반복하여 최종적으로 128bit Whitening을 수행한다.

다음으로 본 논문에서 제안한 핵심 알고리즘인 F 함수로서 F 함수 내부는 확산과 치환 과정이 있으며, 자세한 내용은 2.2절과 2.3절에서 설명 했다. PKC 암호 알고리즘은 전체 2라운드이므로 그림 3의 과정이 정확하게 X축을 기준으로 대칭을 이루고 있다.

복호화 과정은 암호화 과정과 똑 같으며, 복호화 과정에서 라운드 키의 적용은 암호화 과정의 역 순으로 하면 된다.

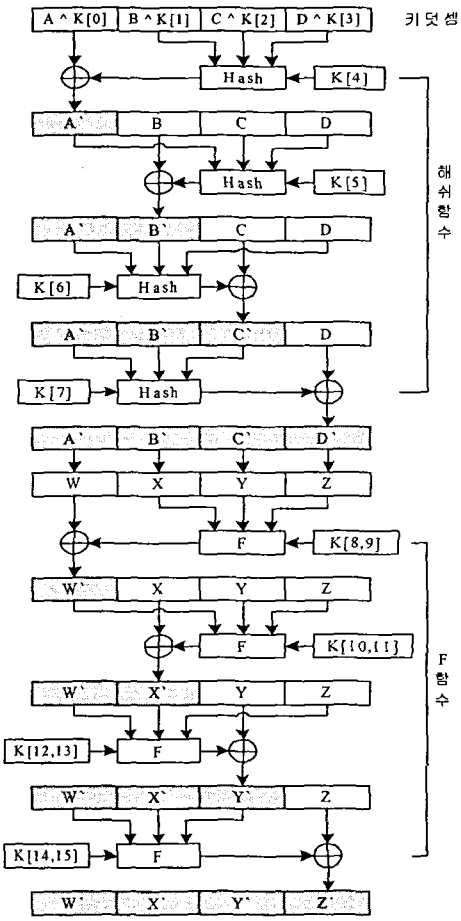

그림 4. 1라운드 암호화 과정

IV. 결론

본 논문에서는 블록 암호 알고리즘 PKC(Pu Kyong Cipher)를 제안했다. PKC 암호 알고리즘은 암호화 과정에서 32bit 단위로 값이 변화하는 변형된 Feistel Network구조이다.

PKC 암호 알고리즘은 128bit 블록 단위로 키는 128, 192, 256bit를 사용 할 수 있도록 구성되었다. 암호/복호는 전체 2라운드로 구성 되었으며, 암호과정에서 8bit S-box 한 개와 XOR연산만으로 간결하게 구성되었으며, 라운드마다 16개의 32bit 라운드 키를 사용하여 전체 라운드 키는 32개이다. 이러한 특성은 제한된 환경에서 수행해야 되는 스마트카드와 같은 응용분야에 많이 활용 될 수 있을 것이다.

제안된 PKC 암호 알고리즘은 소프트웨어로 구현하였으며, 하드웨어로의 구현과 PKC 암호 알고리즘의 안전성을 검정을 위한 블록 암호의 대표적 공격 방법인 차분공격[8](differential cryptanalysis)과 선형공격[9](linear cryptanalysis) 등 에 대한 안전성을 검정하는 것을 향후 과제로 남겨 놓았다.

참고문헌

[1] NIST, http://www.nist.gov/aes/

[2] "Report on the Development of the Advanced Encryption Standard(AES)". http://www.csrc.nist.gov/encryption/aes/round2/r2report.pdf

[3] NESSIE project, New European Schemes for Signatures. Integrity. and Encryption, http://cryptonessie.org/

[4] CRYPTREC REPORT 2000, http://www.ipa.go.jp/security/fy12/report/cryptrec-report2k.pdf

[5] SEED, http://www.kisa.or.kr/seed/

[6] NIST FIPS PUB 197: Advanced Encryption Standard, November 2001.

[7] Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita, Camellia: A 128-bit block cipher suitable for multiple platforms-design and analysis, LNCS 2012, pages 39-56. Sprihger, 2000.

[8] Eli Biham and Adi Shamir, "Differential cryptanalysis of DES-like cryptosystems." Journal of Cryptology 4(1): 3-72, 1991

[9] Mitsuru Matsui, "Linear cryptanalysis method for DES cipher." In Tor Helleseth, editor, Advances in Cryptology-Eurocrypt '93, volume 765 of Lectuer Notes in Computer Science, pages 386-397: Verlag, Berlin, 1994