데이터베이스를 위한 블록 암호 알고리즘 설계

김화영*, 서용무**
*국민대 BIT 대학원 박사과정, **고려대 경영대학

Design of a Block Cipher Algorithm for Database

Kim, Whayoung, Suh, Yongmoo

Kookmin University, Korea University
hyundaisoft@yahoo.co.kr, ymsuh@korea.ac.kr

요 약

최근 기업의 업무처리에 정보시스템의 의존도가 높아짐에 따라, 정보 보안 침해가 더욱 심해지고 있으며, 정보 유출 사례도 증가하고 있다. 이러한 가운데 기업은 중요한 정보를 보호하기 위해서 각종 보안 시스템을 도입하여 사용하고 있으나, 해커들의 불법적인 침입과 기업 내 정보시스템 사용자들에 의한 정보 유출 문제에 매우 취약한 실정이다.

이로 인하여 기업의 중요한 정보가 침해를 받거나 적절하게 보호되지 못하면, 기업의 이미지나 고객의 프라이버시 보호에 매우 큰 손상을 입을 수 있다. 그러므로 데이터베이스의 자료는 변조, 훼손, 유출 등 각종 내·외부의 위협으로부터 안전하게 보호되어야 한다.

이를 위해 기업의 중요 자산인 데이터베이스의 정보보호, 즉 데이터베이스의 정보를 암호화함으로써 정보의 유출 및 침해를 예방하고자, 이에 필요한 효율적이고 안전한 블록 암호 알고리즘의 설계 결과를 제시한다.

1. 서 론

암호 알고리즘은 수 천년 전부터 사용되어 왔다. 그러나 고대 암호는 컴퓨터 기술의 발전에 따라 더 이상 안전성을 보장 받을 수 없게 되었다. 즉 과거 암호 알고리즘은 알고리즘 자체를 비공개로 하는 것을 전제로 사용하여 왔으나, 암호 알고리즘이 노출될 경우 쉽게 해독이 가능하여 암호 알고리즘의 보호가 중요한 보안 매개변수가 되었다. 그러나 암호 알고리즘의 보호는 그 위험이 매우 크고, 다수가 사용하기에는 불편한 점이 많아 매우 비실용적인 방식이다. 그래서 암호 알고리즘을 설계할 시에는 알고리즘의 공개 대신 암호 키를 보안 매개변수로 고려한다.

암호 알고리즘은 안전성과 효율성 측면을 모두 고려하여 적합한 안전도를 갖도록 설계한다. 특히 암호 알고리즘의 안전도는 컴퓨터 계산능력 뿐만 아니라, 알고리즘의 구조적 취약성에 기반을 둔 다양한 분석기법들에 의해 결정됨으로, 암호 알고리즘은 다양한 분석기법을 적용해서 안전성을 입증하여야 신뢰성이 있는 암호 알고리즘으로 사용될 수 있다. 선진국에서는 각종 정보보호

시스템에 대한 안전성의 평가항목과 기준을 공개하고, 이에 근거하여 해당 시스템의 안전성을 평가한다[1].

2. 블록암호 알고리즘 개요

2.1 암호 알고리즘 정의

암호 알고리즘(Cipher Algorithm)이란 비밀성 및 인증을 제공하기 위한 암호학적 알고리즘을 통칭하는 것으로서, 암호화 알고리즘, 전자서명 알고리즘, 해쉬 알고리즘 및 메시지 인증 알고리즘 등이 있다. 그러나 비밀성 기능을 제공하는 알고리즘으로 국한시켜서 좁은 의미의 암호화 알고리즘을 암호 알고리즘이라 하며, 이는 암호화 및 복호화에 사용되는 알고리즘 및 키 스케줄 알고리즘으로 구성된다[1].

암호의 분류는 암호화가 이루어지는 단위의 크기와 키의 형태에 따라 스트림 암호(Stream Cipher)와 블록암호(Block Cipher)로 구분된다. 스트림 암호는 메시지를 각 비트 또는 각 바이트 단위로 암호화하며, 블록암호는 암호화 하려는 메시지를 일정 길이의 블록으로 나누고 각 블록단위로 암호키를 사용, 암호화한다. 또한 사용된 키의 형태로서 비밀키 방식과 공개키 방식으로 나눈다. 비밀키 암호(Secret Key Cipher) 또는 대칭키 암호(Symmetric Key Cipher)는 암호화에 사용된 키와 복호화에 사용된 키가 동일한 암호이다. 공개키 암호 (Public Key Cipher) 또는 비대칭키 암호(Asymmetric Cipher)는 암호화에 사용된 키와 복호화에 사용되는 키가 서로 다른 암호로서 암호화 키는 공개, 복호화 키는 비밀로 한다[2].

현대의 블록 암호 알고리즘은 "혼돈 (Confusion) 및 확산(Diffusion)의 반복 에 의하여 강력한 암호 알고리즘을 설계할 수 있다" 고 하는 Shannon 의 이론을 기반으로 설계된다. 혼돈이론은 암호문 비트들의 통계적 분포가 메시지 비트들의 통계적 분포에 어떻게 의존 되는가를 판단하기 어렵게 만드는 것이고, 확산이론은 메시지의 각 비트 들의 영향이 암호문 비트들에 어떻게 영향을 주는가를 판단하기 어렵게 만든다는 것이다. 따라서 이들 이론에 기초해서 만든 알고리즘들의 비도 (Security Level)는 높아지는 장점이 있는 반면, 단점으로는 블록 단위로 암호화가 이루어지므로 메시지 비트들이 완전한 하나의 블록을 구성한 다음에 암호화의 과정이 이루어 지므로 블록의 크기에 따라 시간이 지연될 수 있다. 블록암호 시스템은 크게 두 부분으로 나눌 수 있는데, 한 부분은 암호 알고리즘 부분, 또 한 부분은 키 생성 부분으로 구성되며, 실제로 암호 알고리즘은 공개되므로 비도는 키 생성 부분에 의존된다[3]. 또한 블록암호 알고리즘은 대치(Substitution)와 치환 (Permutation)을 반복함으로써 높은 비도의 암호를 얻을 수 있으며, 블록 암호 알고리즘에서 중요한 매개변수는 키 길이와 블록의 크기이다. 따라서 이들은 암호시스템의 효율성과 안전성을 크게 좌우한다[1].

2.2 블록암호 알고리즘의 구조

블록암호 알고리즘은 메시지를 처음 라운드의 입력으로 해서 여러 라운드가 반복적으로 수행된 결과로 암호문을 출력하는 반복구조(Iterated Structure) 로 되어 있다. 각 라운드의 구조에 따라 블록암호는 크게 Feistel 구조, SPN (Substitution Permutation Network) 구조 등으로 분류된다.

Feistel 구조는 한 라운드가 입력 블록의 반만이 키와 결합하여 비선형적 대치 단계인 S-box[1]를 통과하여 나머지 반과 배타적 논리합(XOR) 연산을 하고, 변하지 않은 원래의 반과 순서가 바뀌는, 즉 블록의 반은 변하지 않고 다음 라운드의 입력으로 들어가는 구조로 되어 있다. 전체 알고리즘의 라운드 수는 요구되는 보안의 강도와 효율성의 상호 절충적 관계에서 결정된다. 보통 3 라운드 이상의 짝수 라운드로 구성된다. 이러한 Feistel 구조의 특징은 라운드 함수에 관계없이 역변환이 가능하며 (암호화 및 복호화의 과정이 동일), 두 번의 수행으로 블록간의 완전한 확산이 이루어진다. 알고리즘의 수행속도가 매우 빠르고, 하드웨어 및 소프트웨어 구현이 용이하며, 아직 구조상의 문제점이 발견되고 있지 않다는 장점을 지니고 있다[1].

SPN 구조는 혼돈과 확산 효과를 줄 수 있는 라운드 함수를 구성함으로써, 안전하고 실질적인 암호를 설계할 수 있다는 것을 반영한 암호 알고리즘의 구조이다. 즉 이의 한 라운드는 보안 매개변수인 키가 더해진 후 혼돈에 해당하는 비선형적 대치인 S-box 가 확산에 해당하는 비트별 치환(Bitwise Permutation)에 의해 연결되는 구조나 또는 확산 효과를 높이기 위하여 치환을 선형변환으로 대체하는 선형 변환 (Linear Transformation)의 구조로 이루 어진다. 일반적으로 SPN 구조의 한 라운드는 대치, 선형 변환 및 키 덧셈 (Key Addition)이 반복되는 3 단계로 구성된다. 대치 단계는 입력을 몇 개의 작은 블록으로 구분한 후에 각각의 소 블록들에 S-box 의 비선형 변환을 적용하여 출력 값을 얻는 과정이며 혼돈 효과를 주기 위한 단계이다. 선형 변환 단계는 대치 단계의 출력인 각각의 S-box 출력 값을 전체 블록에 골고루 분산시키기 위한 과정으로 확산효과를 주는 구성 요소이다. 그리고 키 덧셈 단계는 서브키를 주입하는 과정이다[4].

1) n x n의 전단사 함수이며, 비선형성을 줌

2.3 블록암호 알고리즘 안전성 분석

블록암호 알고리즘은 대치와 치환을 반복하여 설계되며, 그 구조상 암호 세부논리와 구조 복잡도, 키 특성 및 통계특성 등으로 안전성 분석이 가능하다. 암호세부논리는 암호 알고리즘을 구성하는 라운드 함수와 S-box 이며, 구조 복잡도는 암호 알고리즘의 전체 구조에 따른 복잡도를 말하는 것이다. 키 특성은 생성된 서브키와 대칭키와의 관계를 말하며, 통계특성은 메시지와 암호문, 메시지와 키와 암호문 사이의 통계적 특성 등을 말한다.

현재까지 암호 알고리즘의 안전성에 대한 검증 방법으로서는 입·출력 변화 공격법인 차분 분석법(Differential Cryptanalysis)과 선형 공격법인 선형 분석법(Linear Cryptanalysis) 등이 있다. 이러한 차분 분석법은 1990 년에 Biham 과 Shamir 에 의해서 개발된 암호 알고리즘의 안전성 분석법이며, 또한 선형 분석법은 1993 년에 Matsui 에 의해 개발된 안전성 분석법이다. 블록암호를 설계하기 위해서는 이들 공격법에 대해 안전하게 설계되어야 한다[1].

세부적인 논리 분석에는 S-box 특성 분석, 선형변형의 특성분석 및 라운드 함수의 특성분석을 통해 안전성을 분석한다. 선형변형의 특성분석에는 MDS (Maximal Distance Separable) 행렬 곱셈기를 사용해서 선형변환에 대한 차분 및 선형 분석을 한다. 또한 구조 복잡도 분석에는 S-box 와 선형 변형에 대한 차분 및 선형 분석에 최대특성확률 및 선형근사확률의 상한을 이용한다[4].

암호 알고리즘의 키 스케줄은 대칭 키가 주어지면, 이로부터 라운드 수 만큼의 서브키를 생성하는데 이 들 사이의 연관 관계의 취약성은 연관키 공격(Related Key Attack)의 대상이 된다. 연관키 공격은 서로 다른 두 키 사이의 연관 관계를 알고 있으나 키 자체는 모르는 경우, 각 키로부터 발생된 메시지, 암호문 쌍을 가지고 키를 알아내는 공격 방식인데, 이러한 방법을 사용하여 키 스케줄 특성분석을 통해 안전성에 대한 분석을 한다[1].

블록암호 알고리즘의 통계 특성분석에 의해 안전성을 만족하는지를 결정하기 위하여, 암호 알고리즘의 출력을 8 개의 데이터의 집합(표본군)으로 구성하고, 각각의 데이터 집합들을 16 개 항목의 통계테스트에 적용하여 검정을 실시해서, 암호 알고리즘의 취약성을 분석한다.

블록암호 알고리즘의 안전성을 높이기 위해 부울함수를 사용하면, 부울함수가 만족해야 하는 쇄도 효과(Avalanche Effect)나 또는 SAC(Strict Avalanche Criterion)을 만족시켜야 한다. 쇄도 효과는 주어진 부울함수의 입력 한 비트 가 변하면 출력문의 변하는 비트의 수가 1/2 이어야 한다는 개념이며, 쇄도 효과 를 만족하는 것을 검정하기 위해서 전체 모집단 2^n 개의 메시지(또는 키)x 에 대한 출력문과, 입력 메시지 x 와 i 번째 비트 만 변화시킨 출력문 x' 의 상이한 비트 수가 n/2 개가 되는지를 검정하는 테스트이다. SAC 은 주어진 부울함수의 입력 한 비트가 변하면 출력의 각 비트가 1/2 의 확률로 변해야 한다는 개념이다. 블록암호 알고리즘이 SAC 을 만족함을 검정하기 위해서 모집단 2^n 개의 표본을 조사한다. 즉 2^n 개의 메시지(또는 키)x 에 대한 출력문과 입력메시지 x 와 i 번째 비트만 변화시킨 x' 의 출력문을 배타적 논리합 연산의 결과에 대한 확률이 1/2 이 되면 SAC 을 만족한다고 한다[1].

3. 블록암호 알고리즘 설계 기준

3.1. 알고리즘의 설계기준

블록암호 알고리즘을 설계할 때는 알고리즘의 안전성과 효율성을 동시에 고려해야 하며, 특히 특정분야에 적용 하는 알고리즘일 경우는 그 특성에 중점 을 두고 설계하여야 한다[1].

1) 일반적인 설계기준: Shannon 의 암호 논리인 혼동 및 확산이론을 결합, 암호 알고리즘을 설계하여 원하는 안전도를 갖는 알고리즘을 개발할 수 있는데, 이러한 이론을 바탕으로 블록암호 알고리즘을 설계할 때, 공통적으로 고려되는 기준은 다음과 같다.
   - 키 길이가 충분히 길어야 함
   - 블록 크기가 충분히 커야 함
   - 키와 암호문의 상관 관계가 없어야 함
   - 메시지와 암호문의 상관 관계가 없어야 함
   - 키는 임의적으로 생성되고, 균등 하게 사용되어야 함
   - 암호 알고리즘은 구조적 특이성 이 없어야 함
   - 메시지와 암호문으로부터 키를 추출할 수 없어야 함

3.2. 키 스케줄의 설계기준

키 스케줄은 암호 키가 주어졌을 때, 이로부터 필요한 라운드 수 만큼의 서브키를 생성하는 과정이다[1].

1) 일반적인 설계기준: 키 스케줄을 설계할 때 우선적으로 고려하는 것은 대칭키와 암호문 사이의 상관관계가 존재하지 않아야 하며,

대칭키의 각 비트는 암호문의 모든 비트에 동일한 영향을 주어야 한다. 키 스케줄의 일반적인 설계기준은 다음과 같다.

- 취약키 및 준취약키가 없어야 함
- 암호키는 서브키의 생성에 참여 하는 정도가 균등해야 함
- 보수 특성(Complement Property) 이 없어야 함

2) 연관키 공격에 대한 설계기준: 연관키 공격은 키 사용이 규칙적 으로 변하는 경우나 키 공유 프로토콜에 개입하여 인위적으로 키를 조작할 수 있는 경우에 적용될 수 있다. 이에 대한 키 스케줄 설계기준은 다음과 같다.

- 서브키에서 대칭키를 복원 할 수 없어야 함
- 대칭키의 선형변환으로 서브키의 생성이 없어야 함
- 대칭키의 변화로 서브키의 변화 를 유추할 수 없어야 함
- 대칭키의 모든 비트는 서브 키 발생에 참여하여야 함
- 키 스케줄이 없이 모든 서브키가 독립적인 것은 피해야 함

3.3. 효율성

블록 암호 알고리즘을 설계할 때는 안전성과 효율성이 매우 중요하다. 즉 안전성과 적절한 키 길이, 업무 특성에 적합한 수행속도가 필요하다[1].

블록암호 알고리즘의 속도는 하드웨어 측면이나 소프트웨어 측면에서 빠를수록 좋지만, 더욱 중요한 것은 구현되는 환경에 따라 얼마나 적합한가 하는 것이다. 적용업무 특성에 따라 적합한 것이 가장 바람직하며, 실시간 온 라인 업무의 경우 수행속도는 필수조건이다.

4. 블록 암호 알고리즘 설계

4.1. 블록암호 알고리즘 설계 기준 개요

데이터베이스를 보호하기 위한 암호화 에는 안전성과 효율성, 그리고 타 응용 시스템과의 연동이 잘 될 수 있는 암호 알고리즘이 필요하다. 이에는 블록암호 알고리즘이 가장 적합하다. 본 연구에서 제시하는 블록암호 알고리즘은 Feistel 구조를 기본 구조로 한다. Feistel 구조는 암호화와 복호화의 과정이 같은 알고리즘으로, 복호화 시에는 서브키만 역순으로 적용해서 라운드 함수에 상관 없이 역변환이 가능하고, 알고리즘의 수행속도가 매우 빠르다. 또한 하드웨어 및 소프트웨어적으로 구현이 용이하다는 장점이 있다. 따라서 혼돈과 확산효과가 최소의 라운드 내에서 이루어지도록, 비선형변환과 선형변환을 적절히 조합 하는 구조를 적용해서 암호문의 생성 속도가 빠르고, 외부 공격에도 안전해야 한다. 이를 위한 설계기준을 다음과 같이 결정하였다.

- 128 비트 대칭키 블록 암호
- 128 비트의 키 크기
- 알고리즘의 구조는 Feistel 구조 를 기반으로 함
- 기본적인 라운드 수는 8 라운드 로 하며, 안전성을 높이기 위해 이에 2 라운드 씩 증가시켜 전체 라운드를 가변화해야 함
- 차분공격의 대상인 S-box 대신에 차분분석과 선형 분석에 강한 부울함수를 사용하고, 부울함수 는 SAC 을 만족해야 함
- 키 스케줄은 각각의 입력의 균등성을 보장해야 함

- 모든 연산은 4 바이트 단위로 수행해야 함(다정도 연산이 필요 없으므로 수행 속도가 빠름)

4.2. 암호화 과정

부울함수를 이용하는 대치 단계에서는 간단한 연산으로 메시지를 4 개의 32 비트로 나누고, 입력 단계에서 4 개의 서브키와 배타적 논리합 연산을 한다. 즉, 입력 데이터의 16 바이트(128 비트) 는 처음에 4 개의 4 바이트 씩의 단어로 나눈다. 이들 단어들은 4 개의 서브키와 함께 배타적 논리합 연산을 한다. 2 개의 단어가 라운드함수의 입력으로 사용되고 나머지 2 개의 단어는 앞의 라운드에서 나온 출력과 배타적 논리합 연산을 취한 후, 다음 라운드의 입력으로 사용된다.

라운드함수는 암호학적으로 강한 부울 함수로 구성한다[5].

MDS(Maximal Distance Separable) 행렬 을 이용하는 선형변환 단계에는 앞의 대치 결과를 사용한다. MDS 행렬을 이용한 선형변환 단계의 과정을 거치는 이유는 좋은 확산을 하기 때문이다. 이 선형변환 단계의 결과 벡터는 32 비트 단어(8 비트씩 4 개)로 해석된다. 암호화 및 복호화 모두 배타적 논리합 이후에 한 비트의 오른쪽 회전이 일어난다. 이러한 회전 방법은 MDS 특성을 보존 하도록 선택된다. 변환작업은 하나의 단순한 혼합작업으로 빠르게 수행되며, 이것은 서브 블록들과 키 사이의 혼돈을 위해서 수행된다[6]. 이러한 부울함수를 이용한 대치 단계 및 MDS 행렬을 이용한 선형변환 단계를 거친다. 이후에 마지막 처리단계는 역치환이며 역치환이 이루어 진 후, 4 개의 단어는 암호문을 생성하기 위하여 4 개의 서브키와 또 한번 배타적 논리합이 이루어지고 나서, 128 비트의 암호문이 생성된다.

4.3. 복호화 과정

복호화는 암호화와 같은 알고리즘을 사용하고 암호화의 역순으로 수행하며, 서브키 역시 역순으로 적용한다. 연산에 있어서도 논리합 연산은 논리차 연산 (2 의 보수를 이용한 덧셈)을 한다.

4.4. 키 스케줄 과정

설계 원칙은 먼저 서브키 생성에 있어서 모든 키 비트가 사용되어야 하고, 모든 키 비트는 각 라운드마다 독립적 으로 새로운 키를 생성해야 한다. 또한 취약키가 없어야 하며 라운드변환에서의 대칭성과 라운드 간의 대칭성을 제거 해야 한다. 서브키를 생성하는데 가장 중요한 설계목표는 안전성을 보장하는 것이다.

서브키 생성과정은 128 비트를 32 비트 씩 4 개로 나누고, 이것을 다시 4 개로 나누어 8 비트씩을 대치하여 서브키를 생성한다. 암호키는 16 바이트(128 비트) 로 이것을 4 바이트씩 4 개의 단어로 나누어 처리한다.

첫 번째 단계에서는 32 비트의 입력 데이터를 8 비트씩 4 개로 나누어서 비선형성을 갖는 부울함수를 사용한다.

두 번째 단계에서는 이들을 2 비트씩 회전이동 시킨다. 여기에서 회전이동을 시키는 이유는 이전 라운드의 키 값에 따라 몇 라운드 후에 키 값이 선형변환 될 수도 있어서 이를 방지하기 위함이다.

세 번째 단계에서는 이들을 배타적 논리합(XOR) 연산을 함으로써 서브키를 얻는다. 이러한 구조는 기존의 Feistel 구조와 달리, 이전 라운드의 출력결과가 다음 라운드에 영향을 미치면서, 그와 동시에 입력 블록과 배타적 논리합의 단계에서 변화를 일으켜 복잡도를 증가 시키게 되므로,

비교적 구조는 간단하고 암호 알고리즘의 비도는 높아지게 되는 좋은 구조가 된다[5,6].

4.6. 알고리즘의 안전성 평가

기존의 블록암호 알고리즘에서 사용 하는 S-box 대신에 비선형 특성을 갖는 부울함수를 사용해서 확산성과 차분 및 선형분석에 대해 안전성을 향상시키는데, 이는 암호학적으로 강한 부울함수의 세 가지 성질 중에서, 첫 번째 SAC(Strict Avalanche Criterion)을 만족해야 한다. 이것은 입력비트가 1 비트 차이 날 때에 출력 비트는 최소한 2 비트 이상 차이가 나며, 암호학적으로 함수가 SAC 을 만족 하게 되면, 각 출력 비트는 입력비트의 1 비트가 변하면 1/2 의 확률로 변한다. 두 번째 암호 알고리즘이 비선형이어야 한다는 것은 부울함수의 비선형성(Non-linearity)이 암호학적인 설계에 대해 매우 적합하다는 것이다. 그리고, 세 번째 부울함수가 기본적으로 0 과 1 이 균형(Balance)이 되어야 한다는 것이다 [5].

이러한 세 가지 성질들을 만족하는 부울함수로 구성된 라운드 함수와 MDS 선형변환을 사용함으로써, 차분공격 및 선형공격에 대해 안전성을 제공 한다[7]. 또한 암호 알고리즘에 대한 안전성의 평가는 통계적 검정과 쇄도 효과검정(Avalanche Effect Test)을 통하여 안전성을 확인한다. 이러한 통계적 검정 방법으로는 Frequency Test (빈도 검정), Serial Test(계열 검정), Poker Test(포커 검정) 및 Run Test(런 검정) 등을 사용한다[1].

4.7. 알고리즘의 성능 평가

일반적으로 암호 알고리즘의 성능 평가를 위해서, 암호화 및 복호화 처리 시간과 서브키 생성시간을 측정하는 알고리즘 속도측정 프로그램을 사용하여, 블록 당 사이클(Cycle Per Block)속도로 성능을 비교 평가하게 된다. 이 방법은 클럭(Clock)에 비례하는 속도를 구하기 가 쉬우므로, 서로 다른 알고리즘의 성능을 평가하는 경우에 매우 적합하게 사용하는 측정 방법이다[6]. 이에 따라 키 생성 알고리즘에 의한 서브키 생성 속도를 측정하고, 서브키를 제외한 알고리즘의 계산속도를 측정해서 전체적 인 암호화 및 복호화의 수행속도를 산출한다. 이를 기준으로 다른 암호 알고리즘들인 DES, IDEA, RC5 또는 MARS, RC6, Serpent, Twofish, Rijndael(AES) 들의 키 생성시간과 암호화 및 복호화의 처리속도(Mbps)를 비교하여 성능을 평가 한다[5,6,8].

4. 결 론

정보시스템의 데이터베이스 보호는 매우 중요하므로 기업의 중요한 데이터 베이스의 정보보호를 위해서 효율적이고 안전한 블록암호 알고리즘이 필요하다. 따라서, 본 논문에서는 이에 적합한 알고리즘을 설계하여 제시하였다. 이는 대칭키 암호 알고리즘이며, Feistel 구조를 기반으로 비교적 간단한 연산을 수행하여 속도가 빠르고, 소프트웨어나 하드웨어 구현이 용이하다. 또한 128 비트의 암호키와 가변적인 라운드 수를 적용하며, 데이터베이스의 안전성과 암호화 및 복호화의 처리속도에 따라 라운드 수를 조절할 수 있다. 그러므로 이러한 가변성은 데이터베이스 정보의 특성 및 활용수준에 따라 암호 알고리즘 을 적용할 수 있어서, 정보의 안전성과 효용성을 높일 수 있다는 장점이 있다.

참고문헌

[1] 김승주, 윤선희(2001), *블록암호 알고리즘 설계 및 안전성 분석모델 개발*, 한국정보보호진흥원, 최종연구 개발결과보고서

[2] 육사 수학과(2000). *암호학 개론*, 육군사관학교 수학과, 경문사

[3] 이민섭(1999), *현대 암호학*, 교우사

[4] 최은화, 서창호, 성수학, 류희수 (2002), "DC 와 LC 에 안전한 SPN 구조 암호 알고리즘", *한국정보 처리학회지* 제 9-C 권 제 4 호, pp. 445-452.

[5] 이인실, 심경섭, 김혜정, 신원, 신상욱, 이경헌(1998), "이중 인벌 루션 구조를 지니는 가변길이 블록 암호 알고리즘", *한국멀티미디어 학회 논문집*, 제 1 권 제 1 호, pp. 90-97.

[6] 정혜명, 전문석(2001), "의료정보 보안을 위한 블록 암호 알고리즘의 설계", *한국정보처리학회지* 제 8-C 권 제 3 호, pp. 253-262.

[7] Youssef, A. M., Mister, S., and Tavares, S. E. (1997), "On the design of linear transformations for substitution permutation encryption networks," Workshop in Selected Areas of Cryptography, SAC '97, Workshop record, pp. 40-48.

[8] 이건배, 이병욱(2001), "FPGA 를 이용한 128 비트 암호 알고리즘의 하드웨어 구현", *한국정보처리학회 지* 제 8-C 권 제 3 호, pp. 277-286.