인증기능을 가진 혼합형 암호시스템 설계

이 선 근, 김 영 인, 고 영 욱, 송 재 호, 김 환 용
원광대학교 전자공학과
전화 : 063-850-6740 / 핸드폰 : 016-601-6138

Hybrid Cryptosystem Design with Authentication

Seon-Keun Lee, Young-In Kim, Young-Oog Ko, Jae-Ho Song, Hwan-Yong Kim
Dept. of Electronic Engineering, Wonkwang University
E-mail : caiserrisk@korea.com

Abstract

The importance of protection for information is increasing by the rapid development of information communication and network. Asymmetric cryptosystem is the mainstream in encryption system rather than symmetric cryptosystem by above reasons. But asymmetric cryptosystem is restricted in applying to application fields by the reason it takes more times to process than symmetric cryptosystem. In this paper, the proposed cryptosystem uses an algorithm that combines block cipherment with stream cipherment. Proposed cryptosystem has a high stability in aspect of secret rate by means of transition of key sequence according to the information of plaintext while asymmetric/symmetric cryptosystem conducts encipherment/ decipherment using a fixed key. Consequently, it is very difficult to crack although unauthenticator acquires the key information. So, the proposed encryption system which has a certification function of asymmetric cryptosystem and a processing time equivalent to symmetric cryptosystem will be highly useful to authorize data or exchange important information.

※ 정보통신부에서 지원하는 대학기초연구지원사업으로 수행

I. 서론

급속한 인터넷의 발달과 정보화 사회의 전환으로 인한 정보교환은 현대문명에 커다란 영향을 미친다. 그러나 정보네트워크 발전은 개인정보 누출, 사생활 침해 그리고 금융업 등에 대한 정보개방으로 인한 물질적, 정신적인 피해와 같은 단점을 가지고 있다. 그러므로 네트워크 발전으로 인한 피해를 줄이기 위하여 정보보안에 관한 분야는 네트워크 발달과 더불어 병행되어 발전되고 있다. 이러한 정보보호분야는 알고리즘 개발 및 시스템 개발등과 같이 전문성을 띄며 각 분야별로 발전되는 것이 현재 정보화 사회의 추세이다. 수학적 난이도에 의한 암호화 방식인 비대칭 암호시스템은 네트워크상에서의 키의 관리 및 분배가 용이하다는 장점이 있지만 처리시간이 길다는 단점을 가진다.[1][2][3]

따라서 본 논문에서는 암호화 처리속도가 빠른 대칭형 암호방식을 기준으로 비대칭형 암호방식의 특징을 가지도록 혼합형 암호시스템을 설계하고자 한다.

II. 암호 알고리즘

정보보호를 위한 암호 알고리즘은 그림 1과 같이 대칭형(symmetric, secret)과 비대칭형(asymmetric, public) 암호방식으로 분류된다. 대칭형 암호방식은 구조적인 어려움을 이용하여 암호화를 수행하며 비대칭형 암호방식은 수학적인 해법을 구하기 어렵다는 점을

이용한다. 비대칭형 암호방식은 네트워크 환경에서 키 관리 및 분배가 용이하지만 처리속도가 매우 낮다. 대칭형 암호방식은 처리속도가 매우 빠르지만 네트워크 환경에서 키관리 및 분배가 어렵다는 단점을 가진다. 그러므로 비대칭 암호방식에 비하여 대칭형 암호시스템은 많은 장점에도 불구하고 네트워크 환경에서 사용의 제한을 가지게 된다.

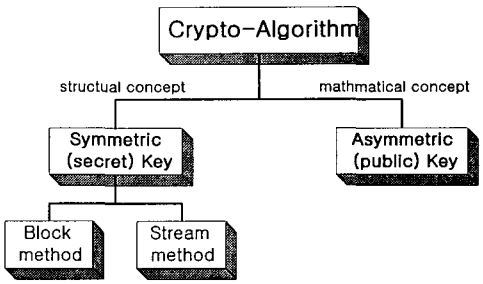

그림 1. 암호 알고리즘의 분류

표 1은 대칭형암호방식과 비대칭형 암호방식의 특징을 비교분석한 표이다. 표에서와 같이 대칭형 암호시스템은 여러 가지 장점에도 불구하고 키분배의 난해성으로 인하여 네트워크 환경에서의 사용이 제한되어 있다. 그러므로 본 논문에서는 네트워크 환경속에서 키분배의 효율성을 가지도록 하기 위하여 대칭형 암호방식에 인증기능을 추가함으로써 대칭형 암호방식의 장점과 비대칭형 암호방식의 장점을 갖춘 혼합형 암호시스템을 제안하고자 한다.[2-6]

표 1. 암호방식에 따른 특징분류

	대칭형 암호방식	비대칭형 암호방식
암호화 원리	구조적	수학적
처리속도	고속	저속
키 분배	난해함	용이함
시스템 구현	용이함	난해함

Ⅲ. 제안된 혼합형 암호시스템

대칭형 암호방식의 구성은 데이터 암호부분과 키 스케쥴 부분으로 분류된다. 데이터 암호부분에서 사용되는 기법은 Feistel 구조와 SPN(substitution and permutation network) 구조를 사용한다. 또한 데이터처리를 어떠한 포맷으로 하느냐에 따라서 스트림 암호방식과 블록 암호방식으로 분류된다. 블록 암호방식과 스트림 암호방식의 가장 큰 차이점은 반복성에 있다. 블록 암호방식은 한 라운드에 대하여 기본적인 안전성을 획득하기 위하여 일정 횟수의 iteration을 수행하게된다. 이와는 다르게 스트림 암호방식은 무한수열에 가까운 LFSR의 주기특성을 가지도록 한다. 그러나 iteration의 증가 또는 LFSR 주기길이의 증가는 대칭형 암호시스템의 효율을 저하시키는 요인이 된다. 그러므로 본 논문에서 제안한 혼합형 암호시스템은 iteration을 사용하지 않고 단일 라운드를 사용하며 LFSR의 stage 길이를 일정한 크기로 설정하고 설정된 LFSR을 다수개 사용함으로써 실제적으로는 매우 긴 주기수열을 생성할 수 있도록 하였다.

제안된 혼합형 암호시스템은 데이터의 재배열(permutation), 치환(substitution), 데이터 암호블록, 키 스케쥴(key schedule)로 구성되어 있다. 데이터 암호화 과정은 128 비트 평문블록을 64 비트씩 2개의 블록으로 분할하고 확장재배열(expansion)을 거친 후 80 비트의 크기를 가지는 혼합형 키(hybrid key : HK)를 사용하여 암호화한다. 기존 블록 암호시스템인 경우 내부적으로 16라운드(16-round)의 암호화 과정을 거치고, 복호화 시에도 암호화에 사용된 동일한 키를 역순으로 사용하여 16라운드의 복호화 과정을 수행한다. 그러나 제안된 혼합형 암호화 시스템은 단일 라운드만을 사용하여 기존의 16 라운드에 해당하는 비도(security level)를 얻기 위하여 혼합형 키와 블록 암호시스템의 비선형 부분인 F 암호함수부분을 보다 더 비선형화 시켰다. 제안된 혼합형 암호시스템은 식 (1)과 같이 128 비트에 대하여 초기치환(initial permutation : IP)을 수행하고 IP 수행결과를 F 암호함수와 혼합 키(HK)를 이용하여 내부연산을 수행한 후 교번하여 출력을 내보내도록 한다.

$$\begin{aligned} L_1 &= R_0 \\ R_1 &= L_0 \oplus f(R_0, HK) \end{aligned} \qquad (1)$$

여기에서 L은 왼쪽 레지스터이며 R은 오른쪽 레지스터이다. 식 (1)은 기존 Feistel 구조와 동일하지만 iteration에 관한 i 파라미터가 없는 대신 이전과 이후에 관한 방정식만 존재한다. 입력 128 비트는 IP를 거친 후 L/R로 좌우 64비트씩 분리된다. 각각 L/R의 좌우로 분리된 64 비트에서 오른쪽 64 비트는 왼쪽으로 이동하며 왼쪽 64 비트는 혼합형 키와 F 암호함수의 연산과정 후 XOR 연산을 통하여 오른쪽으로 이동하게 된다. 이러한 연산을 수행한 후 마지막으로 역 초기치환(inverse initial permutation : IIP)을 수행함으로써 암호화된 데이터를 출력하게된다.

그림 2는 데이터 암호블록의 합성된 회로도이다. 그림 2에서와 같이 데이터 암호블록은 입력 128 비트에

대하여 좌우 레지스터에 64 비트씩 양분하여 데이터를 재포맷팅한 후 F 암호함수를 이용하여 암호화를 수행하게 된다.

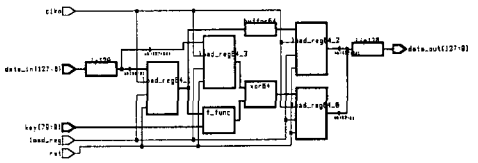

그림 2. 제안된 혼합형 암호시스템의 데이터 암호블록

그림 3은 데이터 암호블록의 모의실험 결과이다. 시스템 동작주파수는 @40Mhz이며 64 비트와 128 비트의 입력값에 대하여 120ns의 처리시간을 가짐을 확인하였다.

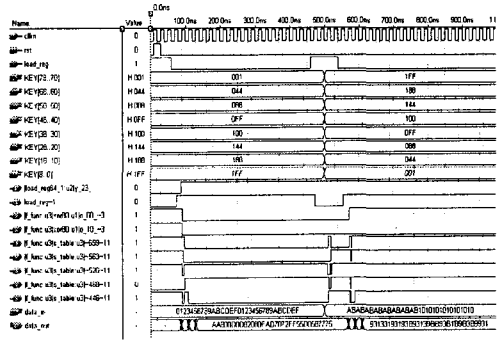

그림 3. 데이터 암호블록에 대한 모의실험 결과

그림 4는 제안된 혼합형 암호시스템에 사용되는 키 스케줄러 블록이다. 키 스케줄러 블록의 기본 구성은 스트림 암호방식이므로 기존 스트림 암호방식과 유사하지만 비도를 증가시키며 비대칭형 암호방식의 특징인 인증기능을 나타내기 위하여 LFSR을 병렬처리로 구성하였고 인증데이터를 얻기 위하여 병렬처리된 LFSR의 LSB와 MSB를 이용하여 인증데이터를 생성하였다.

Mixer의 데이터 $M=\{m_0, m_1, \cdots, m_{62}, m_{63}\}$, replator의 데이터 $R=\{r_0, r_1, \cdots, r_{62}, r_{63}\}$에 대하여 출력값 $key-out$에 대하여 정리하면 식 (2)와 같다.

$$key-out = M(MSB) \oplus R(MSB) \text{ and } M(LSB) \odot R(LSB) \quad (2)$$

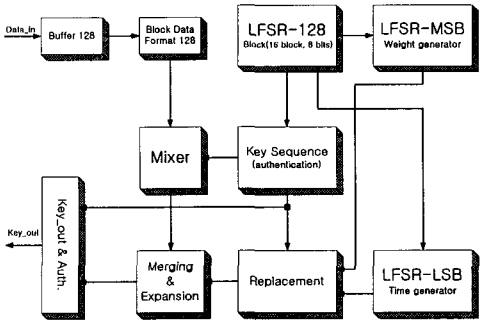

그림 4. 제안된 혼합형 암호시스템의 키스케줄러 블록

그림 4의 전체 출력 $key-out$은 식 (2)의 결과식과 인증기능을 가진 Auth. 데이터를 합한 것으로써 식 (3)과 같다.

$$key-out \text{ and } Auth. = M(MSB) \oplus R(MSB) \text{ and } M(LSB) \odot R(LSB) \text{ and } Auth. \quad (3)$$

여기에서 and는 &와 같이 단순합을 의미하며 ⊕는 비트들끼리의 XOR, ⊙는 비트들끼리의 XNOR 연산을 의미한다. 식 (3)에서 MSB와 LSB를 행렬식으로 표현하기 위하여 사용되는 기본 행렬식은 식 (4)와 같다. 그러므로 식 (3)의 MSB와 LSB를 행렬로 표현하면 식 (5)와 같다.

$$\begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_7 \\ m_8 & m_9 & m_{10} & \cdots & m_{15} \\ & & & \cdot & \\ & & & \cdot & \\ m_{56} & m_{57} & m_{58} & \cdots & m_{63} \end{bmatrix} \text{ XOR, XNOR } \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_7 \\ r_8 & r_9 & r_{10} & \cdots & r_{15} \\ & & & \cdot & \\ & & & \cdot & \\ r_{56} & r_{57} & r_{58} & \cdots & r_{63} \end{bmatrix} \quad (4)$$

$$MSB = \begin{bmatrix} m_{32} & m_{33} & m_{34} & \cdots & m_{39} \\ m_{40} & m_{41} & m_{42} & \cdots & m_{47} \\ & & & \cdot & \\ & & & \cdot & \\ m_{56} & m_{57} & m_{58} & \cdots & m_{63} \end{bmatrix} \oplus \begin{bmatrix} r_{32} & r_{33} & r_{34} & \cdots & r_{39} \\ r_{40} & r_{41} & r_{42} & \cdots & r_{47} \\ & & & \cdot & \\ & & & \cdot & \\ r_{56} & r_{57} & r_{58} & \cdots & r_{63} \end{bmatrix}$$

$$LSB = \begin{bmatrix} m_0 & m_1 & m_2 & \cdots & m_7 \\ m_8 & m_9 & m_{10} & \cdots & m_{15} \\ & & & \cdot & \\ & & & \cdot & \\ m_{24} & m_{25} & m_{26} & \cdots & m_{31} \end{bmatrix} \odot \begin{bmatrix} r_0 & r_1 & r_2 & \cdots & r_7 \\ r_8 & r_9 & r_{10} & \cdots & r_{15} \\ & & & \cdot & \\ & & & \cdot & \\ r_{24} & r_{25} & r_{26} & \cdots & r_{31} \end{bmatrix} \quad (5)$$

식 (5)에서와 같이 XOR, XNOR의 연산이 bit by bit 연산을 기본으로 수행하기 때문에 각 스트림들에 대한 연산은 가중치(weight)가 없는 단순 2진 데이터 연산이다. 이러한 단순연산의 결과, 산출되어지는 데이터 스트림은 사용목적에 따라 별도의 포맷을 수행하게된다. 즉 데이터 암호용으로는 XOR, XNOR연산의 결과

값을 사용하게 된다.

그림 5는 제안된 혼합형 암호알고리즘의 키 스케줄러 블록의 합성된 회로이다. 입력으로는 데이터 암호블록에서 사용되는 원 데이터를 사용하며 출력은 데이터 암호블록의 입력으로 사용되는 HK와 인증용 키 수열로 구별된다.

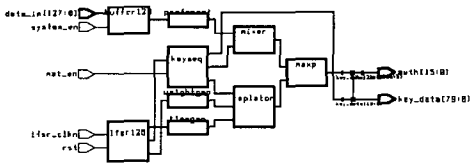

그림 5. 제안된 키스케줄러 합성회로

그림 6은 제안된 키 스케줄러의 모의실험 결과를 나타낸다. 출력데이터는 인증용과 데이터를 암호화하기 위하여 필요한 HK로 구성되어있다. 또한 인증용 데이터는 내부 제어신호인 매트릭스 제어신호에 의하여 제어되므로 인증용 데이터는 사용자가 원하는 시간에 출력할 수 있도록 구성되어 있다.

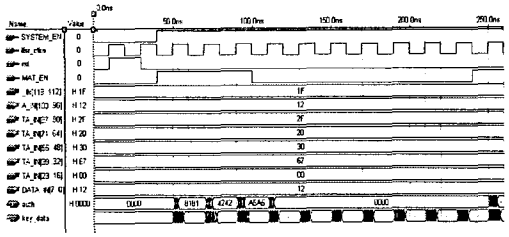

그림 6. 키 스케줄러에 대한 모의실험 결과

Ⅳ. 결론

비대칭형 암호 알고리즘의 가장 큰 장점은 네트워크 환경에서 키의 분배 및 관리가 용이하다는 것이다. 그러나 수학적 난이도의 해법에 의존하는 비대칭형 암호방식은 처리속도가 매우 낮기 때문에 대용량 데이터의 실시간 암호화는 어렵다. 그러므로 본 논문에서는 암호화 처리속도가 빠른 대칭형 암호방식을 기준으로 비대칭형 암호방식의 특징을 가지도록 혼합형 암호시스템을 설계하였다.

제안된 혼합형 암호시스템은 iteration을 가지지 않으며 LFSR의 병렬처리 및 LFSR의 MSB, LSB만을 이용하여 인증용 데이터를 생성하기 때문에 비도 유지 및 DC, LC로부터 더욱 안전하다. 또한 인증용 데이터의 생성은 데이터 블록의 데이터를 사용함으로써 실제적으로 데이터 암호블록과 키 스케줄러 블록은 상호 의존적인 관계를 가진다. 제안된 혼합형 암호시스템의 모의실험 결과 크기면에서는 대칭형 및 비대칭형 암호시스템에 비하여 3배의 증가를 보이나 처리속도면에서는 대칭형 암호방식에 비하여 처리율이 8배 증가함을 확인하였다.

그러므로 본 논문에서 제안된 인증기능을 가진 혼합형 암호시스템은 대용량 데이터의 실시간처리가 가능할 뿐만 아니라 네트워크 환경에서 사용자의 제어에 의하여 비대칭형 암호방식과 같이 사용이 가능하리라 사료된다.

참고문헌

[1] E. Biham, "On the Applicability of Differential Cryptanalysis to Hash Functions", Lecture at EIES Workshop on Cryptographic Hash Functions, Mar. 1992

[2] E. Biham, "On Matsui's Linear Cryptanalysis", Advances in Cryptology-EURO-CRYPT'94 Proceedings, Springer-Verlag, pp. 398-412, 1995

[3] H. Feistel, "Step Code Ciphering System", U.S. Patent #3,798,360, 19 Mar. 1974

[4] E. F. Brickell, J. H. Moore, and M. R. Purtill, "Structure in the S-Boxes of the DES", Advances in Cryptology-CRYPTO'86 Proceeding, Springer-verlag, pp. 3-8, 1987

[5] A. G. Broscius and J. M. Smith, "Exploiting Parallelism in Hardware Implementation of the DES", Advances in Cryptology-CRYPTO'91 Proceeding, Springer-verlag, pp. 367-376, 1992

[6] L. Brown, J. Pieprzyk, and J. Seberry, "Key Scheduling in DES Type Cryptosystems", Advances in Cryptology-CRYPTO'90 Proceeding, Springer-verlag, pp. 221-228, 1990