# 네트워크 환경에 적용하기 위한 대칭형 혼합형 암호시스템 설계에 관한 연구

정우열* · 이선근**

# A Study on the Symmetric Hybrid Cryptosystem Design for Adaptation of Network Environment

Woo-yeol Jeong* · Seon-keun Lee**

요 약

본 논문에서는 정보화 사회와 더불어 네트워크 환경에서 사용하는 여러 시스템들의 정보보안을 위한 보안 시스템을 연구하였다. 이에 따라 복잡성 및 낮은 처리 속도 등의 문제를 해결하기 위하여 블록 및 스트림 방식을 적용한 대칭형 기반 혼합형 암호 시스템을 설계하였다. 인증 기능을 포함한 대칭형 기반 혼합형 암호 시스템은 처리 속도와 계산량이 비대칭형보다 우수한 성능을 가지고 있다. Synopsys 1999.10과 ALTERA MaxPlus 10.1로 시스템을 설계하여 시뮬레이션을 한 결과 제안된 혼합형 암호 시스템은 네트워크 환경에서 정보보안이 필요한 분야에  매우 효율적인 지원과 성능을 제공할 것이다.

ABSTRACT

In this paper, we studied security systems for information security of several systems that use in network environment along with information society. Therefore, we designed symmetry style base mixing style cryptographic system that apply block and stream way to solve problems of complexity and lower processing speed etc.

Symmetry style base mixing style cryptographic system including authentication operation holds performance that the processing speed and the calculation amount are more superior than asymmetry style. Result that design system by Synopsys 1999.10 and ALTERA MaxPlus 10.1 and do simulation, mixing style password system that we propose is that information security offers very efficient assistance and performance in necessary field in network environment.

키워드

Cryptographic System, Symmetric Style, Authentication Operation, Mixing Style

## Ⅰ. 서 론

현대 사회는 수많은 정보를 필요로 하는 고도의 정보화 시대이다. 이런 정보화 물결과 더불어 산업 및 사회 전반에 걸쳐 정보의 중요성은 점점 심화되고 있다. 특히 인터넷과 같은 네트워크를 기반으로 한 대량의 정보 교환이 급속히 증가하는 추세이다. 그리하여 네트워크에 적용하기 위한 성능을 가진 보안 시스템이 매우 중요하다[1].

* 한려대학교 멀티미디어 정보통신공학과
접수일자 : 2007. 07. 15
** 원광대학교 전기전자 및 정보공학부
심사완료일자 : 2007. 08. 23

1990년대까지 사용되어온 대부분의 스트림 암호알고리즘은 LFSR(Linear Feedback Shift Register)을 기본으로 하여 여러가지 비선형적 결합을 통해 주기가 긴 키 스트림을 발생하는 형태가 일반적이었다. 이런 형태는 하드웨어 구현에 용이하고 안전도를 수식적으로 표현할 수 있는 장점이 있지만 소프트웨어의 구현성이 문제가 되어 최근에는 RC4, SEAL3.0, ISAAC 등의 알고리즘과 혼합형 알고리즘이 제안되고 있다[2][3][4].

본 논문에서는 네트워크 환경에 적합하도록 블록 및 스트림 암호알고리즘에 기반을 둔 혼합형 암호알고리즘을 제안하였다. 제안된 혼합형 암호 알고리즘은 네트워크 환경에 대한 구조, 특성, 안전성, 기술 및 다양한 응용부분을 고려하여 연구하였다[2][5].

## II. 기존 대칭형 암호시스템

블록 암호알고리즘은 DES와 같은 형태인 Feistel 구조와 치환(substitution) 및 재배열(permutation)을 반복하여 사용하는 SPN 구조 등으로 구성된다. Feistel 방식은 한 라운드에 평문의 일부만 처리하여 병렬처리 효율이 낮은 반면 라운드 함수 설계의 융통성과 암·복호화 과정이 동일하다는 장점을 가진다. SPN 방식은 한 라운드에서 전체 평문을 암호화하므로 병렬처리가 가능하여 속도가 빠르지만 복호화를 고려하여 암호화 과정을 설계하므로 설계의 폭이 좁다라는 단점을 가진다[3][6][7].

Feistel 방식의 암호화는 반복되는 블록 암호화의 특별한 형태로서 SPN 또는 변환과 라운드 함수를 이용하며 구조는 그림 1과 같다.

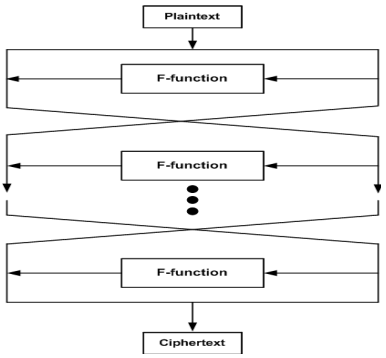


그림 1. Feistel 구조
Fig. 1 Feistel structure

암호화 과정은 먼저, 평문을 우측과 좌측 반씩 두 개 $(L_0, R_0)$로 나누고 라운드 함수 $F$는 서브키$(K_i)$를 우측 반에만 적용하고, $F$ 출력은 좌측의 반과 $XOR$ 연산을 한 후 우측으로 위치가 교환된다. 이때 수행되는 암호화단계는 식 1과 같다.

$$L_i = R_{i-1} \tag{1}$$

$$R_i = L_{i-1} \oplus F(R_{i-1}, K_i)$$

스트림 암호방식은 데이터를 비트 단위로 처리하는 암호방식으로서 주로 유럽을 중심으로 발전되었으며 블록암호에 비하여 암호화의 속도는 빠르고 암호문을 해독하기 힘든 장점이 있다. 블록암호에 비하여 긴 역사를 가지고 있는 스트림암호는 키 스트림 생성기로 사용되는 LFSR, 키 생성방법에 따라 OTP(one time pad), 키 생성이 평문과 암호문에 종속되지 않는 동기 스트림 암호화(synchronous stream cipher) 혹은 KAK(key auto key), 키 생성이 평문과 암호문에 종속되는 자체 동기 스트림 암호화(self synchronous stream cipher, asynchronous stream cipher) 혹은 CTAK(cipher-text auto key)을 근간으로 하지만 최근에는 소프트웨어 구현에 적합한 여러가지 방식이 제안되고 있다. LFSR은 스트림암호에서 의사난수발생기(PRG : pseudo random generator)를 이용하여 수학적으로 분석이 가능한 이진 수열을 효율적으로 발생할 수 있는 장치로 유한체 위에 정의된 선형점화식 수열로 모델링할 수 있으며 수열의 특성은 점화식에 의해 유도되는 특성다항식에 의하여 결정된다.

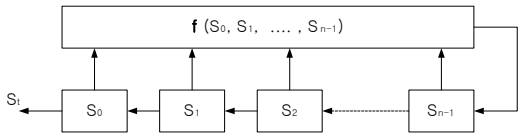


그림 2. LFSR 구조
Fig. 2 LFSR structure

LFSR은 암호뿐아니라 대역확산통신 등에도 많이 활용되고 구현 복잡도가 작아 빠른 속도가 요구되는 곳에 적용할 수 있다. LFSR에서 시프트 레지스터의 일부 셀은 XOR연산을 하고 입력으로 피드백 되어 출력을 생성하는 과정은 그림 2와 같다[3][8][9].유한체 GF(2)={0,1}위에 다

음과 같이 정의된 수열을 식 2로 나타낸다.

$$S_{j+n}=(C_0 S_{j+n-1}+C_1 S_{j+n-2}+\cdots+C_{n-1}S_j), \quad \text{mod}2 \tag{2}$$

여기서, $j\geq 0$, $S_0, S_1, \cdots, S_{n-1}$은 초기치로 정의된다.

## III. 제안된 혼합형 암호시스템

본 논문에서는 정보누출 및 변조를 막고 정보보호 차원에서 하드웨어에 의한 구현을 선택하여 플랫폼의 유·출입부분에서 동작하도록 혼합형 암호알고리즘을 제안하였으며 설계된 암호시스템은 대칭형 암호시스템을 기본으로 비대칭형 암호시스템 특징을 가질 수 있도록 설계하였다.

그림 3은 제안된 혼합형 암호시스템의 데이터 암호 블록으로서 기존 Feistel 구조와 동일한 형태를 취한다.

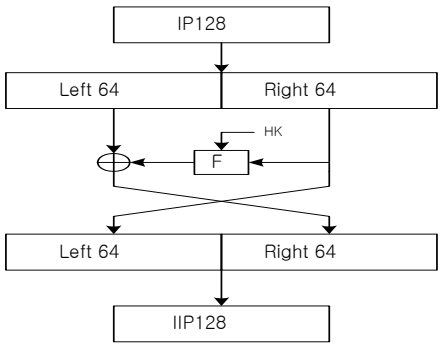


그림 3 Feistel 구조와 SPN 구조
Fig. 3 Feistel structure and SPN structure

식 3은 기본 Feistel 구조와 유사하지만 키생성 및 생성된 키 정보의 형태는 다른 형태를 취하게 된다. 또한 반복에 관한 $i$파라미터가 없는 대신 이전과 이후에 관한 방정식만 존재한다.

$$L_1 = R_0$$
$$R_1 = L_0 \oplus f(R_0, HK) \tag{3}$$

입력 128 비트는 IP를 거친 후 좌우 64비트씩 분리된다. 좌우로 분리된 64 비트는 그림 4와 같이 오른쪽 64 비트는 왼쪽으로 이동하며 왼쪽 64 비트는 혼합형 키와 F 암호함수의 연산과정 후 *XOR* 연산을 수행한 후 오른쪽으로 이동하게된다. 이러한 연산을 수행한 후 역초기치환(IIP : Inverse Initial Permutation)을 수행하여 암호화된 데이터를 출력하게 된다.

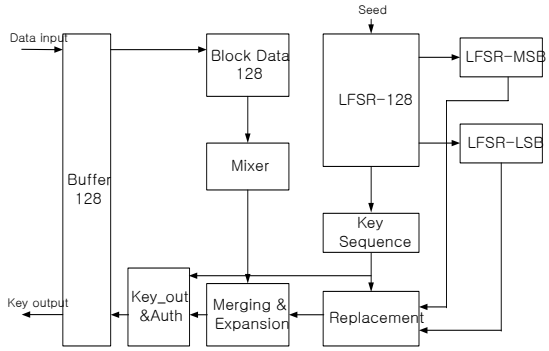


그림 4. 스트림 암호알고리즘의 키 스케쥴러 블록도
Fig. 4 Key scheduler block diagram of stream cryptoalgorithm

일반 블록 암호시스템은 반복을 최소 16회 정도 반복 수행하지만 본 논문에서 사용한 Feistel 구조는 단지 1회의 라운드에 대해서만 수행하도록 한다. 이러한 기능은 F 암호함수에 있다. F 암호함수는 일반적으로 사용되는 키 스케쥴에 의해 발생된 키를 사용하는 것이 아니고 단순 키 기능과 인증 및 비대칭형 개념을 가진 키값을 사용하여 연산을 수행함으로써 비도를 보다 더 높일 수 있다. 키 스케쥴러는 스트림 암호시스템으로 구성되어 있다. 그림 4는 스트림 암호방식을 적용한 키 스케쥴러 블록이다. 키 스케쥴러의 입력으로 특정 키 데이터가 아닌 데이터가 입력으로 사용되므로 그림 5과 같이 Mixer의 입력은 128 비트의 입력데이터와 LFSR로부터 생성된 키 수열이 된다. 이 두가지의 데이터는 16 비트씩 8개의 블록으로 분리되며 각각 XOR 연산을 수행하게된다. 16 비트의 데이터 8 블록들은 XOR 연산 수행 후 메트릭스 치환을 Merging & Expansion 블록의 입력으로 동시에 사용된다.

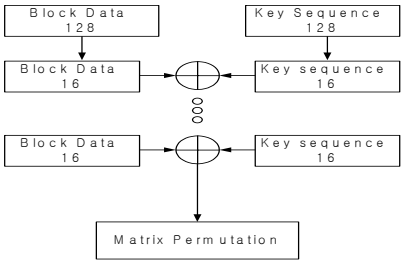


그림 5. Mixer 블록
Fig. 5 Mixer block

메트릭스 치환은 16 비트씩 8 블록이 독립적으로 XOR 연산을 수행하게 되며 행 데이터들에 대한 연산 결과는 열 데이터의 형태로 출력되어진다. 입력데이터의 집합을 I, 키 수열의 집합을 K라 하면 I, K에 대한 표현은 식 4와 같다.

$$I = \{i_0, i_1, i_2, i_3, i_4, i_5, i_6, i_7\}$$
$$K = \{k_0, k_1, k_2, k_3, k_4, k_5, k_6, k_7\} \quad (4)$$

식 4에서 $i$ 와 $k$는 각각 16 비트씩으로 구성된 스트림 데이터들이다. XOR 연산을 수행한 결과를 M이라 하였을 경우 XOR 연산을 수행한 입력 데이터들은 식 5와 같이 표현된다.

$$M = I \oplus K \quad (5)$$

식 5에서 M 역시 16 비트 8 블록이므로 식 6과 같은 수열을 가진다.

$$M = \{m_0, m_1, m_2, m_3, m_4, m_5, m_6, m_7\} \quad (6)$$

식 6에 식 4를 대입하고 연산 수행을 행렬로 표현하여 정리하면 식 7과 같다.

$$M = \begin{bmatrix} ik_{00} & ik_{01} & ik_{02} & ik_{03} & ik_{04} & ik_{05} & ik_{06} & ik_{07} \\ ik_{10} & ik_{11} & ik_{12} & ik_{13} & ik_{14} & ik_{15} & ik_{16} & ik_{17} \\ ik_{20} & ik_{21} & ik_{22} & ik_{23} & ik_{24} & ik_{25} & ik_{26} & ik_{27} \\ ik_{30} & ik_{31} & ik_{32} & ik_{33} & ik_{34} & ik_{35} & ik_{36} & ik_{37} \\ ik_{40} & ik_{41} & ik_{42} & ik_{43} & ik_{44} & ik_{45} & ik_{46} & ik_{47} \\ ik_{50} & ik_{51} & ik_{52} & ik_{53} & ik_{54} & ik_{55} & ik_{56} & ik_{57} \\ ik_{60} & ik_{61} & ik_{62} & ik_{63} & ik_{64} & ik_{65} & ik_{66} & ik_{67} \\ ik_{70} & ik_{71} & ik_{72} & ik_{73} & ik_{74} & ik_{75} & ik_{76} & ik_{77} \end{bmatrix} \quad (7)$$

여기에서 $ik_{32}$는 $i_{32} \oplus k_{32}$의 연산결과임을 나타낸다.

식 7에 대한 매트릭스 치환은 식 8과 같이 전치행렬로 표시되고 연산을 통하여 생성된 64 비트는 Merging & Expansion 모듈의 입력으로 사용된다.

$$M^T = MP = \begin{bmatrix} ik_{00} & ik_{10} & ik_{20} & ik_{30} & ik_{40} & ik_{50} & ik_{60} & ik_{70} \\ ik_{01} & ik_{11} & ik_{21} & ik_{31} & ik_{41} & ik_{51} & ik_{61} & ik_{71} \\ ik_{02} & ik_{12} & ik_{22} & ik_{32} & ik_{42} & ik_{52} & ik_{62} & ik_{72} \\ ik_{03} & ik_{13} & ik_{23} & ik_{33} & ik_{43} & ik_{53} & ik_{63} & ik_{73} \\ ik_{04} & ik_{14} & ik_{24} & ik_{34} & ik_{44} & ik_{54} & ik_{64} & ik_{74} \\ ik_{05} & ik_{15} & ik_{25} & ik_{35} & ik_{45} & ik_{55} & ik_{65} & ik_{75} \\ ik_{06} & ik_{16} & ik_{26} & ik_{36} & ik_{46} & ik_{56} & ik_{66} & ik_{76} \\ ik_{07} & ik_{17} & ik_{27} & ik_{37} & ik_{47} & ik_{57} & ik_{67} & ik_{77} \end{bmatrix} \quad (8)$$

$$\begin{aligned} z_0 + k_0 &= g_1 s_1 + g_4 s_6 + g_8 s_8 \\ z_1 + k_1 &= g_1 k_0 + g_4 s_5 + g_8 s_7 \\ z_2 + k_2 &= g_1 k_1 + g_4 s_4 + g_8 s_6 \\ z_3 + k_3 &= g_1 k_2 + g_4 s_3 + g_8 s_5 \end{aligned} \quad (9)$$

$$\begin{aligned} z_4 + z_4 &= g_1 k_3 + g_4 s_2 + g_8 s_4 \\ z_5 + k_5 &= g_1 k_4 + g_4 s_1 + g_8 s_3 \\ z_6 + k_6 &= g_1 k_5 + g_4 k_0 + g_8 s_2 \\ z_7 + k_7 &= g_1 k_6 + g_4 k_1 + g_8 s_1 \end{aligned}$$

$$\begin{aligned} z_8 + z_8 &= g_1 k_7 + g_4 k_2 + g_8 k_0 \\ z_9 + k_9 &= g_1 k_8 + g_4 k_3 + g_8 k_1 \\ z_{10} + k_{10} &= g_1 k_9 + g_4 k_4 + g_8 k_2 \\ z_{11} + k_{11} &= g_1 k_{10} + g_4 k_5 + g_8 k_3 \end{aligned}$$

$$\begin{aligned} z_{12} + z_{12} &= g_1 k_{11} + g_4 k_6 + g_8 k_4 \\ z_{13} + k_{13} &= g_1 k_{12} + g_4 k_7 + g_8 k_5 \\ z_{14} + k_{14} &= g_1 k_{13} + g_4 k_8 + g_8 k_6 \\ z_{15} + k_{15} &= g_1 k_{14} + g_4 k_9 + g_8 k_7 \end{aligned}$$

식 9는 생성다항식과 LFSR 의사난수발생기의 초기조건을 이용하여 단순화시킨 방정식이다. 키 수열은 그림 6과 같다. LFSR 128로부터 출력되는 데이터를 재포맷하는 블록으로써 LFSR-128의 출력을 기본으로 하여 새로운 수열로 설정한 후 새로 형성된 수열을 이용하여 혼합형 암호시스템의 비밀키와 공개키로써 사용한다. 즉 키 수열에서 출력은 혼합형 암호시스템에서 비밀키와 공개키에 관한 정보를 모두 포함하게 된다. 그림 6과 같이 키 수열 입력은 LFSR 128로부터의 의사난수 입력이다. 입력되는 의사난수는 8단에 불과한 비도가 낮을수도 있는 데이터들이다. 그러므로 이러한 비도 저하를 방지하며 혼합형 암호시스템의 비밀키와 공개키로 사용할 키 값을 추출하기 위하여 재포맷을 수행한다. 이러한 재포맷 과정은 치환연산을 사용하지 않고 데이터들에 대한 매트릭스 포맷을 수행함으로써 외부로 혼합형 암호시스템에 대한 자료가 노출되어도 그 값

을 파악하기 어렵도록 하였다.

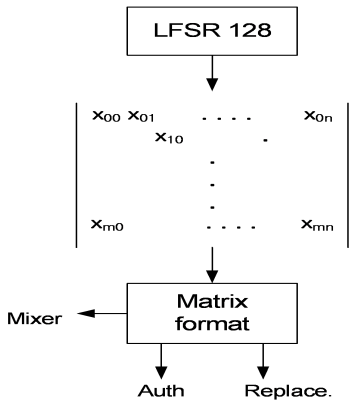


그림 6. 키 수열 블록
Fig. 6 Key Series Block

대칭형 암호시스템은 K라는 비밀키를 송신자와 수신자가 동일하게 보유해야 한다. 사용자 수가 증가하거나 암호화하여 전송해야할 데이터의 블록크기가 매우 길어진다면 대칭형 암호시스템의 장점인 처리속도 효율이 크게 저하될 것이다. 또한 암호화된 데이터와 키 데이터량이 증가하여 암호화 성능도 저하된다. 본 논문에서는 기존 혼함형 암호알고리즘과 같이 암호화를 수행하지 않고 그림 6과 같이 암호화하고자 하는 데이터를 키 데이터로 사용하며 키 수열 생성은 LFSR과 키 재포맷팅을 이용하여 블록 데이터의 크기에 상관없이 키 수열을 생성할 수 있도록 하였다.

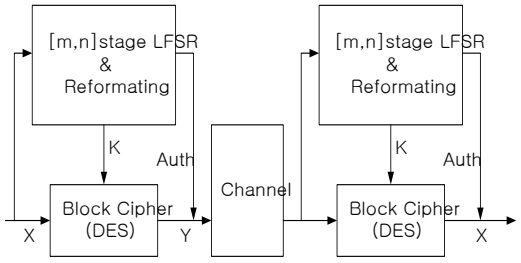


그림 7. 제안된 암호시스템
Fig. 7 The password system which is proposed

그림 7과 같이 입력 데이터 X에 대하여 키 스케줄러에서 X에 의한 키 값 K를 생성하고 생성된 키값 K와 입력데이터 X를 이용하여 암호화된 출력값 Y를 생성한다. 또한 키 스케줄러에서는 암호화용 키값과 인증용 수열을 동시에 생성하며 이 두값은 전송로(channel)를 통하여 수신자에게 전송된다.

제안된 암호시스템은 기존 혼합형 암호시스템과는 매우 큰 차이를 가진다. 기존 혼합형 암호시스템은 비도를 증가시키기 위해서는 LFSR의 주기를 길게 해야하며 블록 데이터의 값도 증가되는 단점을 가지고 있다. 또한 데이터와 키값을 별도로 사용함으로써 암호화에 필요한 자원관리를 항상 염두에 두고 있어야 한다. 특히 키값은 고정된 값으로 존재하게 되므로 비인가자의 키값 획득은 암호화의 필요성을 무력화시키는 중요한 요인으로 작용된다. 그러나 본 논문에서 제안된 그림 8과 같은 네트워크 환경에 적합한 암호시스템은 유입되는 정보데이터를 기반으로 암호화에 사용되는 키값을 생성하므로 데이터에 따라서 암호화 패턴이 변화되는 장점을 가지고 있다.

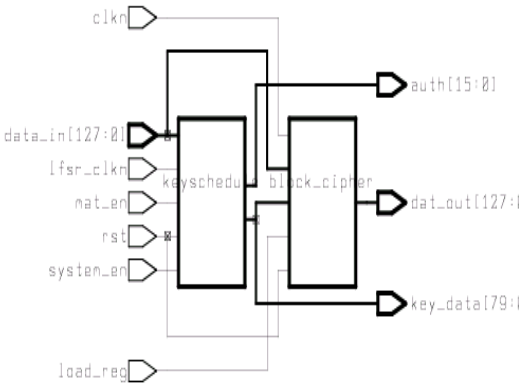


그림 8. 제안된 혼합형 암호시스템의 전체 블록도
Fig. 8 영문 제목 넣어주세요!

대칭형 암호방식에 비대칭형 암호개념을 적용한 새로운 혼합형 암호알고리즘은  입력신호를 평문과 키 데이터로 동시에 사용하고 생성된 암호문은 인증용 정보를 산출하며 단일 라운드로 비도와 F 암호 함수부분을 더욱더 비선형화 시켰다.

본 논문에서 제안된 새로운 혼합형 암호알고리즘을 이용하여 스마트 카드의 암호시스템에 적용한 결과를 표 1로 나타내었다. 제안된 알고리즘과 기존의 알고리즘을 6개의 파라미터(parameter)로 비교해 보면 다음과 같이 성능이 향상됨을 확인하였다.

표 1. 기존 암호 알고리즘과 새로운 혼합형 암호 알고리즘의 비교
Table 1. Comparison of general & new hybrid crypto algorithm

| 알고리즘 | DES | 3DES | SEED | AES | hybrid |
|---|---|---|---|---|---|
| 구 조 | Feistel | Feistel | Feistel | SPN | Feistel &SPN |
| 라운드수 | 16 | 48 | 16 | 10 | 1 |
| 데이터 길이 (bit) | 64 | 64 | 128 | 128/192 /256 | 128 |
| 키 길이 (bit) | 56 | 112/ 168 | 128 | 128 | 80 |
| 시스템 속도 (MHz) | 80 | 80 | 40 | 33 | 40 |
| 데이터 처리율 (Mbps) | 50 | 25 | 251 | ㄴ256 | 640 |

## Ⅳ. 결 론

정보보호 시스템은 비도는 높고 처리속도가 빠른 암호시스템에 기반을 두는 것으로 비대칭키 암호알고리즘으로는 RSA, ECC등이 있으며 대칭키 암호알고리즘으로는 DES, SEED, AES등이 있다. 기존 블록 암호방식은 16 라운드의 암호화와 복호화를 수행하고 스트림 암호방식은 안전성을 획득하기 위하여 무한수열에 가까운 LFSR의 주기특성을 가지도록 한다. 그러나 반복의 증가 또는 LFSR 주기 길이의 증가는 비밀키 암호시스템의 효율을 저하시키는 요인이 된다.

본 논문에서는 네트워크 환경에 적합한 블록 및 스트림 암호알고리즘에 대하여 고찰하였다. 그리고 제안한 혼합형 암호알고리즘을 네트워크 환경에 대한 구조, 특성, 안전성, 관련 표준, 기술 및 다양한 응용부분을 고려하여 연구하였다.

대칭형 암호방식에 비대칭형 암호개념을 적용한 새로운 혼합형 암호알고리즘은 데이터 재배열, 치환, 데이터 암호블록, 키 스케쥴러로 구성되어 있고, 데이터 암호화 과정은 128 비트 평문 블록을 64 비트씩 2개의 블록으로 분할하고 확장을 거친 후 80 비트 크기의 혼합형 키를 사용하여 암호화하였다. 또한, 단일 라운드만을 사용하고도 기존 16 라운드에 해당하는 비도를 얻도록 혼합형 키와 블록 암호시스템의 비선형 부분인 F 암호 함수부분을 더욱더 비선형화 시켰다. 제안한 혼합형 암호알고리즘을 이용하여 네트워크 환경에 적합한 암호시스템을 Synopsys ver 1999.10으로 설계하였고 40MHz의 시스템 속도환경에서 Altera MAX+PlusⅡ툴로 모의실험 및 검증한 결과 단일 라운드로 640Mbps의 데이터 처리율을 확인하였다. 따라서, 제안된 혼합형 알고리즘을 네트워크 환경의 암호시스템에 적용할 경우 입력 신호를 평문 및 키 데이터로 사용할 수 있고 암호문은 비인가자에게 인증용으로 적용되며 기존시스템과 호환성이 용이하여 하드웨어 설계와 보안기능 및 처리속도를 향상시킬 수 있다고 생각된다.

## 참고 문헌


[1] W. Stallings, “Cryptography and Network Security”, Prentice Hall, 1998.

[2] B. Schneier, “Applied Cryptography : Protocols, Algorithms and Source Code in C”, John Wiley Sons, Inc., New York, USA, 1994.

[3] D. R. Stinson, “Cryptography Theory and Practice”, Chapman & Hall/CRC, 2002.

[4] R. Jenkins Jr., "ISAAC", Fast Software Encryption'96, Springer-Verlag, pp.41-49, 1996.

[5] D. Coppersmoth, D. Vagner, b. Schneier and J. Kelsy, "Cryptanalysis of Twoprime", Fast Software Encryption '98, Springer-Verlag, pp.32-48, 1998.

[6] T. Siegenthaler, "Decrypting a Class of Stream Ciphers Using Ciphertext Only", IEEE Trans. on Computer, Vol. C-34, No. 1, pp.81-85, Jan. 1985.

[7] VISA Open Platform Overview, 1999.

[8] H. Miyano, AMethod to Estimate the Number of Ciphertext Pairs for Differential Cryptanalysis, Abstracts of ASIACRYPT91, 1991.

[9] R. Rueppel, "Stream Ciphers", Contemporary Cryptology: The science of Infor. Integrity, New York, IEEE Pres, pp.65-134, 1991.

## 저자 소개

**정우열(Woo-yeol Jeong)**

현재 : 한려대학교 멀티미디어 정보통신공학과교수
※주관심분야 : 이동통신시스템, 암호시스템, VLSI 설계

**이선근(Seon-keun Lee)**

현재 : 원광대학교 전기전자 및 정보공학부 전임강사
※주관심분야 : 이동통신시스템, 암호시스템, VLSI 설계