韓國컴퓨터情報學會 論文誌
第15卷 第2號, 2010. 2.

2010-15-2-5-4

통신에서의 SEED와 스트림 암호 알고리즘의 비교 분석

안 인 수*

SEED and Stream cipher algorithm comparison and analysis on the communication

In-soo Ahn*

요 약

인터넷과 네트워크 등 통신의 급속한 발달로 디지털 정보화 사회가 점점 고도화되고 다양한 서비스를 제공받고 있는 상황이지만 심각한 보안 위협에 노출되어 있다. 이와 같은 정보 보호 시장의 특성에 따라 보안 기술의 적용 환경이나 분야에 적합하고, 중요한 정보를 보다 안전하게 보호하기 위한 암호 기술의 연구가 더욱 절실히 요구된다. 통신에서 정보 보호를 위한 암호화 기술 중에서 암호화 키와 복호화 키가 같은 대칭키 암호 알고리즘은 변환 방법에 따라 블록 암호 알고리즘과 스트림 암호 알고리즘으로 구분된다. 본 연구에서는 제안한 SEED와 스트림 암호 알고리즘의 안전성과 신뢰성을 검증하고 통신 환경에서의 적용 가능성을 확인하고자 한다. 이것은 다양한 통신 환경 조건에 따른 적합한 암호 알고리즘의 선택과 적용으로 안전한 정보 교류가 이루어질 수 있도록 하는데 기여할 수 있을 것이다.

Abstract

Society of digital information becomes gradually advancement, and it is a situation offered various service, but it is exposed to a serious security threat by a fast development of communication such as the internet and a network. There is required a research of technical encryption to protect more safely important information. And we require research for application of security technology in environment or a field to be based on a characteristics of market of an information security. The symmetric key cipher algorithm has same encryption key and decryption key. It is categorized to Block and Stream cipher algorithm according to conversion ways. This study inspects safety and reliability of proposed SEED, Stream cipher algorithm. And it confirms possibility of application on the communication environments. This can contribute to transact information safely by application of suitable cipher algorithm along various communication environmental conditions.

▸ Keyword : Stream Cipher Algorithm, Block Cipher Algorithm, SEED, LFSR

• 제1저자 : 안인수
• 투고일 : 2010. 02. 01, 심사일 : 2010. 02. 06, 게재확정일 : 2010. 02. 22.
* 경인여자대학(Kyungin Women's College)
※ 본 연구는 2008년도 경인여자대학 교내연구지원비에 의해 수행되었음.

Ⅰ. 서 론

인터넷과 네트워크의 급속한 발달로 디지털 정보 사회가 고도화 되고 있다. 또한, 컴퓨터 네트워크의 개방화에 따라 안전성과 신뢰성 확보를 위한 암호 기술의 중요성은 정보 보호를 위해 더욱 더 강조되고 있다. 이와 같은 정보 보호 시장의 성장에 발맞춰 암호 기술의 적용 환경이나 적용 분야에 적합한 지속적인 개발이 요구된다.

다양한 환경에 적합한 암호 기술의 개발로 응용 분야를 확대하고 정보 보호 제품의 개발 비용을 최대한 줄여 국내의 독자적인 기술을 확보한다면 외국 기술의 유입을 최대한 방지할 수 있을 것이다. 정보 보호를 위한 암호화 기술 중에서 암호화 키와 복호화 키가 같은 대칭키 암호 알고리즘은 변환 방법에 따라 블록 암호 알고리즘과 스트림 암호 알고리즘으로 구분된다[1].

본 연구는 개선된 SEED 암호 알고리즘과 스트림 암호 알고리즘을 제안하고, 그것의 안전성과 신뢰성을 검증하여 통신 환경에서의 적용 가능성을 확인하고자 한다.

Ⅱ. 연구 동향

암호 기술은 암호화 기술과 암호 프로토콜 기술로 나뉘고 암호 분야는 크게 블록 암호와 스트림 암호, 공개키 암호 등으로 분류된다. 블록 암호는 1980년대 후반에 특허가 크게 증가한 이후 꾸준한 성장세를 보이고 있으며, 최근 정보 보호 기술의 핵심이 되고 있는 공개키 암호와 스트림 암호 분야도 꾸준한 증가세를 나타내고 있다.

암호 알고리즘의 경우 이미 호주, 일본 등 세계 각국들은 자국의 암호화 알고리즘을 개발하여 정보 보호 차원의 암호화를 실용화하고 있으며, 국내에서도 국제적인 흐름에 따라 한국정보보호센터가 1998년 10월에 초안을 개발하여 안전성과 성능이 검증된 128비트 블록 암호 표준안으로 SEED 블록 암호 알고리즘을 제안하였다[2]. 현재 SEED는 다양한 정보의 안전한 저장이나 교류를 위해 활발히 사용되고 있다.

스트림 암호 알고리즘은 1970년대부터 유럽을 중심으로 발달하였으며, 평문의 연속된 이진 표현으로 암호문을 생성하는 구조이다. 스트림 암호 시스템에서는 키 이진 수열을 어떻게 발생시켜 평문과 결합시키느냐가 암호 시스템의 안전도에 직접적인 영향을 미친다.

암호문을 복호화 하여 평문을 찾을 때 키 스트림과 암호문 사이에 동기가 필요한 동기식 스트림 암호 시스템은 암호문에 들어 있는 키 스트림과 암호문의 독립성으로 인하여 허가받지 않은 사용자에게 정보가 노출될 가능성은 적다. 이 같은 특성으로 스트림 암호 알고리즘은 수학적으로 안전성을 엄밀하게 분석할 수 있어 군사 및 외교용으로 널리 사용되고 있다. 최근 유비쿼터스 컴퓨팅 환경이나 초고속 환경 등에서의 정보 보호를 위해 암호 기술이 많이 요구되어 이에 맞는 암호 알고리즘의 개발이 활발하다.

현재 고속 소프트웨어 환경에 적합하고 128비트 이상의 안전도를 갖는 암호 기술이 개발되어 있다. 블록 암호 알고리즘에서 키 비트의 증가는 해당 알고리즘의 복호화를 어렵게 만드는 요소 중의 하나이다. 대칭키 알고리즘 중에 키 80비트인 블록 암호 알고리즘과 스트림 암호 알고리즘의 성능 평가에서는 안전성에 취약점 없음이 보고되어 있다[3]. 그러나 암호 알고리즘 개발 당시에 안전성이 검증되었다 하더라도 암호 알고리즘에 대한 해킹의 우려는 항상 잠재되어 있고, 모든 통신 환경을 전제로 한 것은 아니므로 이에 대한 보완과 미리 강화된 암호 알고리즘의 개발로 만약의 상황에 대처할 수 있도록 지속적인 연구가 수반되어야 한다.

Ⅲ. 연구 내용 및 방법

대칭키 암호 알고리즘은 비밀키 또는 단일키 암호 알고리즘이라고도 하며, 동일한 키에 의해 암호화와 복호화 과정을 수행한다. 대칭키 암호 알고리즘은 암호화 함수가 주어졌을 때 암호화 함수의 역함수인 복호 함수가 쉽게 얻어진다. 따라서 대칭 암호계에서는 암호화 함수가 알려지면 암호화된 정보는 쉽게 복호화 될 수 있으므로 암호화 함수의 비밀이 유지되어야 한다. 이러한 비밀 유지를 위해 비밀키를 적용하며 그렇기 때문에 비밀키 암호 알고리즘이라고도 한다. 대칭키 암호 알고리즘은 입력되는 데이터 방법에 따라 블록 암호 알고리즘과 스트림 암호 알고리즘으로 구분된다.

본 연구에서는 제안한 SEED 암호 알고리즘의 키 생성 알고리즘에서 라운드 키에 대한 암호키의 영향, Weak Key, Semi-Weak Key, Complementation Property, Equivalent Key 등의 검증과 제안한 스트림 암호 알고리즘의 주기와 선형 복잡도, 상관성 등의 검증을 통해 안전성 또는 신뢰성 여부와 통신 환경에서의 적용 가능성을 확인한다.

1. 블록 암호와 스트림 암호 알고리즘

블록 암호는 암호 알고리즘과 키 블록으로 구성되며, 일반적으로 암호 알고리즘은 개방되므로 비도는 키 블록에 의존한다. 블록 암호 알고리즘에 있어서 출력 블록의 각 비트는 입력 블록과 키의 모든 비트에 의존하여 결정되도록 암호화하며, 입·출력 블록의 크기는 구현상의 효율성과 보안상의 안전성에 크게 의존한다.

블록 암호 알고리즘은 암호화하려는 메시지를 일정 길이의 블록 단위로 나누어 입력 비트 블록을 결정하고, 입력 비트와 같은 크기의 출력 비트로 변형하는 암호 알고리즘에 의하여 암호화 및 복호화가 수행된다. 이 경우 암호화의 블록 길이가 정해져 있으므로 기호의 삽입이나 제거가 불가능하고, 다양한 운영 방식에 의하여 Shannon이 발표한 논문의 혼돈(confusion)과 확산(diffusion)의 이론을 적용하여 설계할 수 있는 것이 장점이다. 이것은 암호문 비트들의 통계적 분포가 평문 비트들의 통계적 분포에 어떻게 의존되는가를 판단하기 어렵게 만드는 혼돈 이론과 평문의 각 비트들의 영향이 암호문 비트들에 어떻게 영향을 주는가를 판단하기 어렵게 만드는 확산 이론을 기반으로 암호의 비도를 높게 할 수 있다. 단점은 블록 단위로 암호화하기 때문에 평문 비트들을 완전히 하나의 블록으로 구성한 다음에 암호화가 이루어지므로 블록 크기에 따라 암호화 과정이 지연되며, 오류의 전파 가능성이 상대적으로 크다. 블록 암호 알고리즘의 대표적인 예로는 Feistel 알고리즘, NDS 시스템, Lucifer 시스템, DES, FEAL, IDEA 등이 있고, 국내에서는 1999년 2월에 최종 제안된 SEED가 있다. 다양한 알고리즘 개발과 함께 DES, AES, RSA, CC 등 많은 알고리즘에 대해 공격 방안과 대응 기술도 연구되었다.[4~6]

스트림 암호 알고리즘은 비밀키 암호 알고리즘의 하나로 블록의 크기를 1로 하여 블록마다 각각 다른 키를 사용하여 암호문을 생성한다. 그러므로 스트림 암호를 사용하기 위해서는 평문($P_1, P_2, P_3, \ldots, P_n$)과 같은 크기의 키 스트림 ($K_1, K_2, K_3, \ldots, K_n$)이 필요하다(그림 1 참조).

스트림 암호는 각 단위마다 각각 다른 비밀키로 암호화를 수행하므로 오류가 일어난 곳 이외에 다른 곳에는 영향을 미치지 않는다[7]. 또한 블록의 크기가 1이고 이전 블록의 결과와 상관없이 암·복호화 수행이 가능하므로 블록 암호에 비해 메모리에 저장할 필요가 없어 고속의 암·복호화가 가능하다.

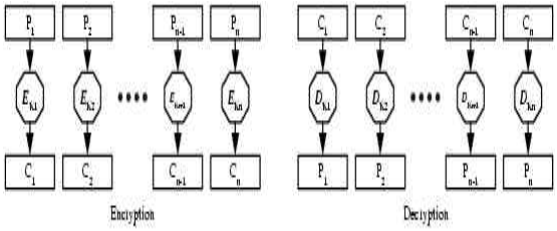

그림 1. 스트림 암호 알고리즘의 구조
Fig. 1. Structure of stream cipher algorithm

스트림 암호는 블록의 단위를 1비트로 하여 평문과 같은 길이의 키 스트림을 XOR하여 암호문을 생성한다. 암·복호화에 사용될 키 스트림은 사용자 사이에 미리 교환되는 경우도 있으나, 일반적으로 사용자의 비밀키로부터 키 스트림 생성 함수를 이용하여 필요한 길이만큼을 생성한다. 키 스트림 생성 함수를 예측 가능한 함수로 사용하면 스트림 암호는 쉽게 해독이 가능하다.

이동 통신에서 정보의 침해나 불법 사용이 문제점으로 대두되면서 정보 보호의 중요성은 더욱 크게 부각되고 있다. 유럽의 디지털 이동 통신 시스템인 GSM 방식과 미국의 CDMA 방식에서는 정보 보호를 위해 사용자 인증과 데이터를 암호화하여 전송하는 암호 시스템을 사용하고 있다[8]. 그러나 GSM의 스트림 암호 알고리즘의 하나인 A5 알고리즘은 구조의 불안성이 있음이 문제시 되었고 해독 가능함이 증명되었다[9]. CDMA 방식의 스트림 암호 알고리즘인 ORXY도 해독이 가능함이 확인되었고, 단거리 이동 단말기에서 주로 사용하는 Bluetooth의 스트림 암호 알고리즘인 E0도 해독되어[10], 이동 통신 단말기 상에서 전송되는 데이터 내용이 노출되어 기밀성 보장이 없음을 확인되었다.

2. 개선된 SEED 암호 알고리즘

제안한 알고리즘은 192비트의 입력 데이터와 256비트의 키 입력 데이터를 갖는다. 256비트의 키 입력 데이터는 128씩 각각 키 생성 과정을 수행하며, 생성된 값을 서로 XOR하여 각 라운드의 최종 킷값을 생성한다. 이렇게 생성된 키는 192비트 입력 데이터와 함께 라운드부의 입력값으로 전달되어 라운드 함수를 수행한다.

라운드 함수부는 표준 SEED 블록 암호 알고리즘와 같은 함수 수행을 적용하였다. 192비트의 입력 데이터를 64비트씩 세 부분의 L_0 (191 downto 128), M_0 (127 downto 64), R_0 (63 downto 0)으로 나눈 것이며, 각 라운드에서의 연산과 32 라운드의 함수 수행을 거쳐 최종 암호화된 데이터를 출력한

다. 복호화는 암호화된 데이터를 입력으로 하여 역으로 수행한다. 라운드 수를 i 라 하고, 라운드 함수는 식 (1)과 같이 나타낼 수 있다.

$$\begin{aligned} L_i &= M_{i-1}\, M_i = R_{i-1} \oplus F(M_{i-1}, K_i) \\ R_i &= L_{i-1} \oplus F(M_{i-1} K_i), \ (1 \le i \le 31) \\ L_{32} &= R_{31} \oplus F(M_{31}, K_{32}), \ (i = 32) \\ M_{32} &= L_{31} \oplus F(M_{31}, K_{32}), \ R_{32} = M_{31} \end{aligned} \quad \cdots\cdots\cdots (1)$$

암호 키 입력 데이터는 256비트로 이것을 각각 64비트씩 네 부분으로 나누어 교대로 8비트씩 좌·우로 회전 이동한 후, 결과의 4워드(word)들에 대한 산술 연산과 G함수를 적용하여 회전키를 생성한다. (그림 2)는 제안한 알고리즘의 전체 구조이다.

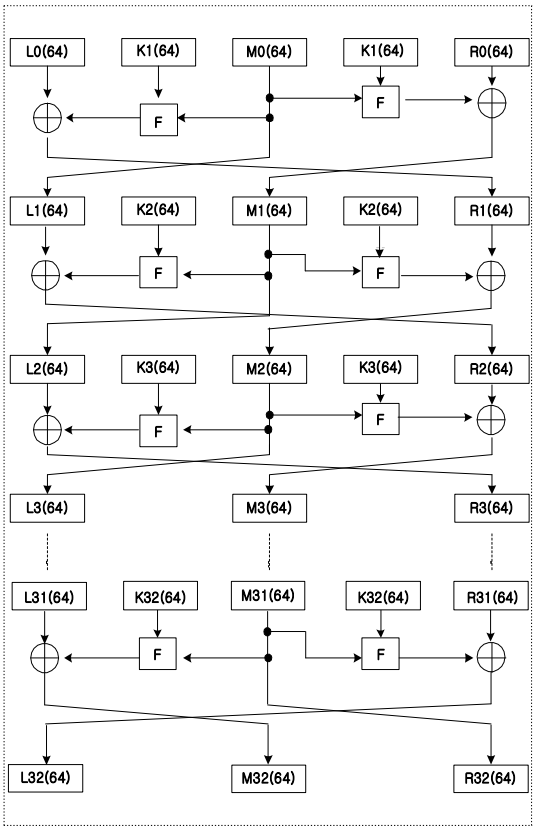

그림 2. 제안한 알고리즘의 전체 구조
Fig. 2. Overall structure of the proposed algorithm

제안한 알고리즘의 키 생성 알고리즘은 G 함수와 64비트 회전 이동을 사용하여 64비트의 서브키 32개를 생성한다. G 함수는 비선형성이 높은 S-box와 평문의 통계적 특성이 암호문에 골고루 반영될 수 있도록 충분한 확산성을 가지는 bit-wise 연산으로 이루어진다. 이때 256비트의 각 비트가 모든 라운드의 서브키에 골고루 반영된다. 라운드 키 상수의 생성이나 라운드별 서브키 생성 방법이 각 라운드마다 동일하게 수행되므로 자원의 효율적인 이용이 가능하다.

키 생성 알고리즘에 의해 생성된 32개의 라운드 키는 데이터패스부에 전달되어 라운드 함수에 적용되고, 총 32회의 라운드 함수를 수행한다. 192비트의 입력 데이터는 32개의 라운드 키와 함께 암·복호화를 위한 데이터패스부의 입력으로 전달된다. 입력된 데이터와 라운드 키는 F 함수를 적용하여 라운드의 출력값을 생성하고, F 함수는 G 함수를 포함하여 수행한다. 생성된 라운드 출력값은 두 번째 라운드 키와 함께 2라운드의 라운드 입력값으로 전달되고 F 함수를 적용한 후, 또 다른 라운드 출력값을 생성한다. 이와 같은 과정을 총 32회 반복한다. 복호화 시에는 역으로 라운드 키 32에서 1까지 순차적으로 전달된다.

3. 개선된 스트림 암호 알고리즘

스트림 암호 알고리즘은 일반적으로 선형 쉬프트 레지스터를 이용하여 구현된다. 낮은 선형 복잡도를 갖는 기존의 스트림 암호 알고리즘의 비도 향상을 위해 제안한 알고리즘은 개선된 비선형 처리부를 구성하였다.

일반적으로 선형 쉬프트 레지스터는 주로 최대 주기를 갖는 m-LFSR(maximum length Linear Feedback Shift Register)을 비선형적으로 결합한 비선형 이진 수열 발생기를 기본으로 한다. 그러나 m-LFSR은 특성 다항식이 갖는 n단 선형 쉬프트 레지스터에 의해 생성된 이진 수열은 연속적인 2n개의 항을 가지고 전체 수열을 발생하므로 연속적인 비트만 알면 피드백 탭(feedback tap)[11]을 알아낼 수 있어 암호 시스템용으로는 적합하지 않다. 따라서 이러한 점을 보완하기 위해 비선형적 특성의 함수 적용으로 선형 복잡도를 높이는 방법을 사용한다.

본 논문에서는 4개의 선형 쉬프트 레지스터 LFSR을 비선형 결합 함수를 적용하여 구성한다. (그림 3)은 제안하는 알고리즘의 전체 블록도를 나타낸 것으로 (그림 3)을 살펴보면, 키 값을 각 선형 쉬프트 레지스터에 저장시킨 4개의 LFSR의 출력값을 비선형 처리부의 비선형 결합 함수인 Fi의 입력값으로 전달한다. 비선형 처리부는 비선형 결합 함수와 S-box로 구성되며, 키 스케줄 입력부에서 생성된 키 값을 비선형 처리부

의 비선형 결합 함수 F_i를 적용한다. 이렇게 비선형 결합 함수를 적용하여 생성된 출력값은 다음 단계의 S-box를 거쳐 비선형 처리부에서의 출력값을 생성한다. 비선형 처리부에서 출력값은 전송 대상 데이터와 XOR 하여 암호화된 최종 출력값을 생성하여 이것을 무선 채널 상에서 전송하게 된다.

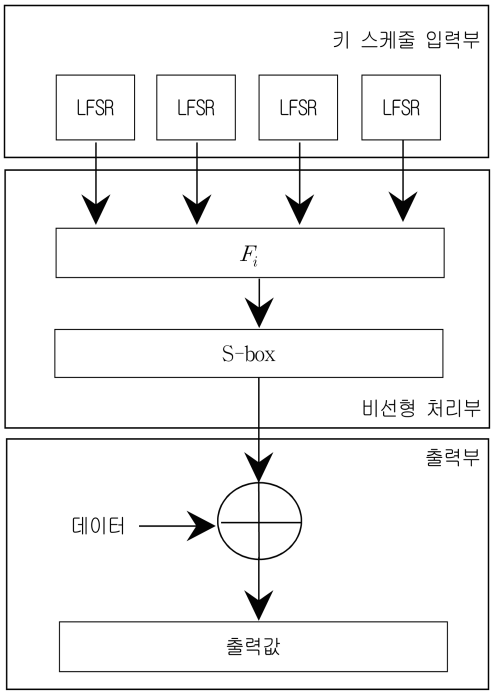

그림 3. 제안한 알고리즘의 블록도
Fig. 3. Block diagram of the proposed algorithm

Ⅳ. 분석 및 고찰

1. 개선된 SEED 암호 알고리즘 분석

키 생성 알고리즘을 이용한 공격 방법은 암호의 응용 범위에 제약을 주는 요인이 될 수 있다. 암호키는 키 생성 알고리즘에 의해 암·복호화 과정에서 필요로 하는 서브키와 라운드 키의 형태로 변환된다. 변환된 키 생성 알고리즘을 분석함으로써 암호 알고리즘 해독을 시도하는데 일반적으로 Weak Key, Semi-Weak Key, Complementation Property, Equivalent Key 성질을 이용한다.

제안된 키 생성 알고리즘에 의해 생성된 암호키의 영향 등을 분석하기 위해 다음 사항들을 고려한다. 우선, 다음 사항에서는 평문을 P, 암호키 K, 암호문 C, 암호화 알고리즘 E, 복호화 알고리즘 D로 정의한다. 참고로 암호키 K를 사용하여 i번째 라운드의 서브키를 생성하는데 사용되는 레지스터 변수를 A, B, C, D, E, F, G, H로 정의한다.

1.1 라운드 키에 대한 암호키의 영향

키 생성 알고리즘은 전체 256비트의 암호키가 각각 128비트씩 L 블록과 R 블록으로 나뉘어 균등하게 32개의 키 상수 KC_i를 적용하여 32비트 단위의 연산이 이루어지도록 한다.

1.2 Weak Key

Weak Key는 동일한 키를 이용하여 두 번의 암호화를 수행하면 다시 평문을 얻을 수 있는 키로 식 (2)과 같이 정의할 수 있다.

$$P = E_K(E_K(P)) \quad \cdots\cdots (2)$$

키 생성 알고리즘에 사용되는 32비트
A, B, C, D, E, F, G, H를
$A = a_3a_2a_1a_0$, $B = b_3b_2b_1b_0$, $C = c_3c_2c_1c_0$,
$D = d_3d_2d_1d_0$, $E = e_3e_2e_1e_0$, $F = f_3f_2f_1f_0$,
$G = g_3g_2g_1g_0$, $H = h_3h_2h_1h_0$와 같이 1바이트 단위 32개로 나타 수 있다.

제안한 키 생성 알고리즘에서 Weak Key가 존재한다면 1라운드와 16라운드, 8라운드와 9라운드, 17라운드와 32라운드, 24라운드와 25라운드 키가 동일해야 한다. 본 알고리즘에서 적용한 G 함수는 일대일 대응으로
$KC_{1,0} - KC_{16,0} \neq KC_{9,1} - KC_{8,1}$,
$KC_{17,0} - KC_{32,0} \neq KC_{25,1} - KC_{24,1}$이므로
Weak Key는 존재하지 않음을 확인할 수 있다.

1.3 Semi-Weak Key

Semi-Weak Key는 서로 다른 키를 이용하여 두 번의 암호화를 수행하면, 다시 평문을 얻을 수 있는 키를 말하는 것으로 식 (3)과 같이 정의할 수 있다.

$$P = E_{K_1}(E_{K_2}(P)) \quad \cdots\cdots (3)$$

Semi-Weak Key가 존재한다면 K_1을 이용한 i, $16+i$ 라운드 키와 K_2를 이용한 $17-i$, $33-i$ 라운드 키가 동일해야 한다.

$KC_{16}-KC_1 \neq KC_8-KC_9$

$\neq KC_{32}-KC_{17} \neq KC_{24}-KC_{25}$이므로 모순이다. 따라서 Semi-Weak Key는 존재하지 않음을 확인할 수 있다. R 블록의 경우도 위와 같은 과정을 수행하면 Semi-Weak Key가 존재하지 않음을 알 수 있다.

1.4 Complementation Property

Complementation Property는 키 K에 대한 평문 P의 암호문이 K의 bit-wise complement인 K'에 의한 P'의 암호문의 bit-wise complement와 동일한 특성을 말한다.

예를 들어 암호 알고리즘 E가 Complementation Property를 가지는 경우, $C=E_K(P)$이면,

$C'=E_{K'}(P')$ 이다. 서브키와 평문 입력이 XOR 되는 구조에서 Complementation Property가 성립하기 위해서는 서브키 생성 과정과 암호 함수가 bit-wise shifting, bit-wise permutation, XOR 등의 비트별 연산으로 해야 한다.

반면에 S-box, 모듈러 가산 등을 사용하는 경우에는 일반적으로 Complementation Property를 만족하지 않는다. 1라운드 함수의 서브키 입력과 함수의 입력을 살펴봄으로써 Complementation Property의 성립 여부를 조사하게 된다. 1라운드의 서브키 64비트는 평문 입력 64비트와 XOR 되며, $a \oplus b = a' \oplus b'$이고, $a \oplus b' = a' \oplus b = (a \oplus b)'$ 이므로 Complementation Property를 만족하기 위해서는 1라운드 서브키가 Complementation Property를 만족해야 한다. 그러나 제안한 키 생성 알고리즘에서의 서브키들은 비선형적인 출력을 생성하는 S-box를 사용하므로, K'에 의해 생성된 서브키들은 K에 생성되는 서브키들과는 무관하다. 또한, F 함수 내에서 모듈러 연산이 사용되므로 Complementation Property를 만족하지 않는다.

1.5 Equivalent Key

Equivalent Key는 평문 P를 서로 다른 두 암호의 키로 암호화하였을 때 생성되는 두 암호의 암호문이 동일한 경우 사용되는 두 암호의 키를 말한다. 이것은 서로 다른 두 암호의 키 K, K^*에 대해 식 (4)와 같이 나타낼 수 있다.

$$C = E_K(P) = E_{K^*}(P) \quad \cdots\cdots (4)$$

K와 K^*가 Equivalent Key가 되기 위해서는 각각에 의해 생성된 모든 서브키가 동일해야 한다. 만약, K와 K^*가 Equivalent Key 라면, $K_1=K^*_1$, $K_9=K^*_9$로 $(A,B,C,D)=(B_9,A_9,D_9,C_9)$이므로 식 (5)를 만족한다.

$$\begin{aligned} &A+C=A^*+C^*,\ B-D=B^*-D^* \\ &B+D=B^*+D^*,\ A-C=A^*-C^* \end{aligned} \quad \cdots\cdots (5)$$

$A'=(A-A^*), B'=(B-B^*), C'=(C-C^*),$ $D'=(D-D^*)$라 가정하면, 식 (6)과 같이 나타낼 수 있다.

$$\begin{aligned} &A'+C'=0,\ A'-C'=0 \\ &B'+D'=0,\ B'-D'=0 \end{aligned} \quad \cdots\cdots (6)$$

$A'=B'=C'=D'=0$이므로, 서로 다른 Equivale-nt Key Pair는 존재하지 않는다. 위의 알고리즘은 음성통신에서 데이터의 암·복호화를 확인하였으며[12], 향후 이동 통신 환경에서의 암·복호화된 데이터 전송에 대한 고찰이 요구된다.

2. 개선된 스트림 암호 알고리즘 분석

키 스케줄 입력부의 선형 레지스터 값을 비선형 처리부의 비선형 결합 함수 F_i를 적용하여 얻어진 값은 다음 단계의 S-box로 거친다. 최종 출력된 키 수열 사이클은 다양한 키 수열 사이클을 제공하므로 출력값으로부터 키 값을 얻어내려는 Dawson과 같은 다양한 상관 공격에 안전하다.

2.1 주기

제안한 스트림 암호 알고리즘의 주기는 4개의 서로 소인 선형 쉬프트 레지스터를 사용하므로 식 (7)과 같이 성립함을 알 수 있다.(P_i : 각각의 주기, L_i : 선형 쉬프트 레지스터의 각 길이, P_e : 제안한 알고리즘 주기)

$$P_i=(2^{L_i}-1)$$
$$P_e=P_1\cdot P_2\cdot P_3\cdot P_4=\prod_{i=1}^{4}P_i \quad \cdots\cdots (7)$$

비선형 결합 함수의 주기인 P_f와 S-box 출력의 주기인 P_s, XOR된 데이터의 주기 P_{xor}는 식 (8)과 같이 알고리즘의 구성에 상관없이 동일한 주기를 가짐을 확인할 수 있다.

$$P_f=P_s=P_e\prod_{i=1}^{4}P_i \quad \cdots\cdots (8)$$

2.2 선형 복잡도

비선형 처리부의 출력값이 서로 소인 경우 본 알고리즘의 선형 복잡도는 두 개의 서로 소인 선형 쉬프트 레지스터가 비선형 결합 함수에 입력되는 경우와 동일하므로 이때의 선형 복잡도 LC_e는 식 (9)와 같이 나타낼 수 있다.

$$LC_e\approx\prod_{i=1}^{4}P_i \quad \cdots\cdots (9)$$

출력을 각 선형 쉬프트 레지스터에 대해 XOR 하는 LC_{xor}와 비선형 결합 함수 선형 복잡도 LC_f, S-box의 선형 복잡도 LC_s는 식 (10)과 같음을 확인할 수 있다.

$$LC_{xor}\approx\prod_{i=1}^{4}P_i,\quad LC_f\approx\prod_{i=1}^{4}P_i, \qquad (10)$$
$$LC_s\approx\prod_{i=1}^{4}P_i,\quad LC_e\approx\prod_{i=1}^{4}P_i$$

$LC_{xor}, LC_f, LC_s, LC_e$는 각 선형 쉬프트 레지스터의 주기의 곱으로 높은 선형 복잡도 특성을 나타내어 비도 요소 중 하나인 높은 선형 복잡도의 조건을 만족하므로 데이터의 기밀성을 보장한다 할 수 있다.

2.3 상관성

비선형 결합 함수는 주어진 함수를 임의의 시간대별로 사용하여 비선형 결합 함수의 변환 요청이 있으면 이전의 결합 함수를 새로운 비선형 결합 함수로 변경한다. 또한, S-box도 시간대별로 해당 S-box만을 사용하고 임의의 시간대에 요청이 있으면 새로운 S-box를 적용하여 변환하는 방법을 사용한다.

기존 모델의 상관성 C와 Dawson 공격에 대한 안전성 비교 결과에 의해 m개의 비선형 결합 함수와 n개의 S-box를 사용하여 키 수열 사이클을 확장하면 Dawson의 공격에 안전하게 되어 이동 통신상에서 전송되는 데이터를 안전하게 보호할 수 있으므로 비도가 향상된다. 즉, 초기값인 세션 키가 변경될 경우, S-box와 비선형 결합 함수를 새롭게 변경하도록 하기 때문에 키 수열 $K=k_0, k_1, k_2, \cdots, k_n$가 새롭게 변경되면 새로운 키 수열 사이클 상에서 암호문이 생성되므로 현재의 암호문 C와 과거의 암호문 C'의 상관성을 유추하기가 더욱 어렵게 된다.

V. 결론

본 논문에서는 개선된 SEED 암호 알고리즘과 스트림 암호 알고리즘을 제안하고 그것의 안전성과 신뢰성에 대한 검증을 통해 통신 환경에서의 적용 가능성을 고찰하였다.

개선된 SEED 암호 알고리즘은 192비트 입력 데이터와 256비트 키 입력으로 16라운드 함수를 수행한다. 키 생성 알고리즘에서의 성능 분석에서 라운드 키에 대한 암호키의 영향, Weak Key, Semi-Weak Key, Complementation Property, Equivalent Key의 검증을 통해 킷값의 안전성을 확인하였다. 그러나 이것은 음성 통신을 대상으로 데이터의 암·복호화가 수행됨을 확인하였으나 이동 통신 환경에서의 안전성은 미흡하다. 스트림 암호 알고리즘은 일반적으로 선형 쉬프트 레지스터를 이용하여 구현한다. 제안한 스트림 암호 알고리즘은 낮은 선형 복잡도를 갖는 기존의 스트림 암호 알고리즘을 개선한 것으로 비선형 처리부를 거쳐 암호화된 데이터를 안전하게 전송할 수 있도록 한다. 제안된 알고리즘의 주기와 선형 복잡도, 상관성을 고찰함으로써 안전성을 확인할 수 있었다. 또한, 기존 모델의 상관성 C와 Dawson 공격에 대한 안전성 비교 결과에 의해 m개의 비선형 결합 함수와 n개의 S-box를 사용하여 키 수열 사이클을 확장하면 Dawson의 공격에 안전하게 되어 이동 통신상에서 전송되는 데이터를 안전하게 보호할 수 있다.

본 연구는 다양한 통신 환경 조건에 따른 적합한 암호 알고리즘의 선택과 적용으로 안전한 정보 교류가 이루어질 수 있도록 하는데 기여할 수 있으며, 향후 제안한 SEED 알고리즘이 이동 통신 환경에서 어떻게 적용될 수 있는지에 대한 연구와 고찰이 요구된다.

참고문헌

[1] 이선근, "유무선 네트워크 환경에 적합한 VCR 암호 시스템 설계에 관한 연구," 한국컴퓨터정보학회논문지, 제14권, 제7호, 66쪽, 2009년 7월.

[2] "*128비트 블록 암호 알고리즘(SEED) 개발 및 분석 보고서*," 한국정보보호센터, 1998년 12월.

[3] "*차세대 암호시스템 개발에 관한 연구*," 국가보안기술연구소, 2쪽, 2006년.

[4] M. Akkar and C. Giraud, "An Implementation of DES and AES, Secure against Some Attacks," CHES 2001, LNCS 2162, pp. 309-318, Springer-Verlag, 2001.

[5] S. Yen, "Amplified Differential Power Cryptanalysis on Rijdael Implementation with Exponentially Fewer Power Traces," ACISP 2003, LNCS 2727, pp. 106-117, Springer-Verlag, 2003.

[6] R. Bevan and E. Knudsen, "Ways to Enhance Differential Power Analysis," ICISC 2002, LNCS 2578, pp. 327-342, Springer-Verlag, 2003.

[7] 김종엽, 김환용, "PRN을 이용한 키 스케쥴러 블록암호시스템 설계에 관한 연구," 한국컴퓨터정보학회논문지, 제8권, 제2호, 114쪽, 2003년 6월.

[8] R. Nichols and P. Leakkas, "*Wireless Security Models, Threats, and Solutions*," McGraw Hill, 2002.

[9] Chi-Chun Lo and Yu-Jen Chen, "Stream ciphers for GSM Networks," IEEE International Conference on Communications(ICC) 2000, Vol. 1, No. 18-22, pp. 80-84, Jun. 2000.

[10] M. Hermelin, "Cryptographic Properties of the Bluetooth Combination Generator," Helsinki University of Technology Department of Engineering Physics and Mathematics, 28th Feb. 2000.

[11] A. Kanso, "*Clock-Controlled Generators*," Ph.D. Thesis, Royal Holloway and Bedford New College University of London, 1999.

[12] 안인수, "통신시스템을 위한 암호 알고리즘에 관한 연구," 대한전자공학회논문지, 제43권, 제2호, 16-21쪽, 2006년 6월.

저 자 소 개

안 인 수

1992년 : 국민대학교 전자공학과 공학사

1994년 : 국민대학교 전자공학과 공학석사

2002년 : 국민대학교 전자공학과 공학박사

현재 : 경인여자대학 정보미디어학부 부교수

관심분야 : 암호, 회로 시스템