論文
9-2-1

암호화 원리 및 도구 분석에 관한 연구
Research about encryption principle and tool analysis

남 태 희(Tae Hee Nam)[1)]

요 약

본 논문은 plaintext 및 image encryption을 위해 암호화의 원리를 이론적으로 고찰하였다. 암호화 방법에 있어서 과거 암호화 방법은 단순히 문자를 치환(permutation cipher 또는 transposition cipher) 하거나 이동하는 방법으로 이용되어 왔으나, 현재는 key stream generator를 이용하는 방식이 이용되고 있다. 즉 평문에 key를 생성하여 암호 및 해독한다. 즉 key를 생성하는 방법에 따라서 암호화의 체계가 달라지는 것이다. 따라서 본 논문에서는 암호화의 원리 및 도구를 고찰하고, 대표적으로, XOR 연산자 및 key stream generator 가정하에서 암호화 원리를 고찰하였다.

ABSTRACT

In this study, investigated principle of encryption theoretically for plaintext and image encryption. Encryption method does character substitution(permutation cipher or transposition cipher) simply past in encryption method or had been used by method to move, but mode to use key stream generator present is used. That is, creating key in plaintext and encryption/decryption. That is, system of encryption according to method that create key changes. Investigate principle and a tool of encryption in treatise that see therefore, and representatively, investigated encryption principle under XOR operator and key stream generator condition.

논문 접수 : 2008. 4. 22.
심사 완료 : 2008. 5. 6.

1) 동주대학 의료기공학과 교수

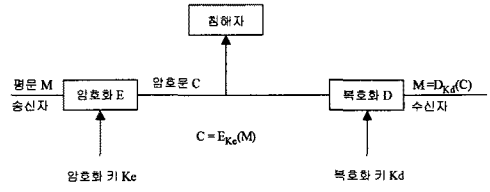
※ 본 논문은 교내 학술 연구 조성비 지원에 의해 연구되었음.

1. 서론

암호화(encryption)는 작게는 개인의 소중한 비밀을 보호하고 크게는 국가적 중요 정보를 보호하는데 목적이 있다. 원래 암호화은 군사적인 목적으로 주로 사용되어 왔으나, 최근에는 컴퓨터 및 통신의 발달인해 주요 정보가 해커의 표적이 되고 있다. 이러한 암호화는 인터넷의 발달로 개인 및 단체의 전자상거래 등이 보편화되어 가고 있어 암호화의 사용은 더욱 중요도를 높여 가고 있다. 국내에서는 인터넷 등 통신의 발달에 비해 암호화의 활용도가 다른 나라에 비해 뒤떨어진 상태이다. 이와 관련해서 국내 주요 정보 보호에 대해 최소한의 장치가 바로 자료의 암호화인 것이다. 따라서 자료 송수신시에 발생되는 여러 가지의 문제를 해결하기 위해 암호화의 중요성이 요구되고 있다. 주로 암호화에 관련되는 주요 문서(plaintext)는 군사기밀, 정부정책문서, 사업기밀 등이 암호 시스템의 주요 암호화 대상 요인이다. 현재 암호화는 상업적으로 많이 이용되면서 정보화 사회의 상거래의 핵심 요소인 전자화폐(electronic money, digital money, cyber cash, virtual currency), 전자송금, 전자 인증(*authentication*), 디지털 서명(*digital* signature), 전자 지갑 실현 및 전자 상거래(electronic business) 등의 신뢰성과 비밀성을 제공해주는 방법으로 암호 방식이 활용되고 있으므로 그 중요도는 실로 다양하다고 사료된다. 본 논문은 여러 암호화 방식들을 개념적으로 파악하고 향후 효과적인 암호화 방식의 새로운 방향을 제시하고자한다.

2. 암호화 방식

암호는 정보 내용과 정보 운송자 사이에 존재하는 다양성을 이용해서 정보 내용과 정보 운송자 사이의 대응 관계를 제삼자에게 비밀로 하여 정보를 교환하는 방법을 의미한다. 물론 정보를 교환하려는 송신자와 수신자는 정보 내용과 정보 운송자 사이의 대응 관계, 즉 key를 사전에 알고 있어야 한다. 따라서 정보를 교환하려는 송신자와 수신자가 공유하고 있는 key가 제삼자에게 알려지지 않도록 유의해야 한다. 일반적인 정보 암호 방식은 다음 fig. 1과 같다.



[그림 1] 암호 방식 구성
[Fig. 1] encryption method structure

평문(plaintext)은 송신자가 수신자에게 전달하려는 정보 내용으로 누구나 그 의미를 알 수 있는 정보이다. 송신자는 평문 M을 암호화 key K_e와 암호 알고리듬(encryption algorithm)을 적용시켜 암호문(ciphertext) C를 생성하여 수신자에게 전달한다.

$$C = E_{k_e}(M) \text{ ----------(1)}$$

암호문(ciphertext) C는 전송 상태에서 그 내용을 알 수 없는 데이터이다. 수신자는 송신자가 전송한 암호문 C를 수신하여 복호화 key와 복호화 알고리듬(decryption algorithm)을 적용시켜 송신자가 전송하려는 평문 M을 복원할 수 있다.

$$M = D_{k_d}(C) \text{ ----------(2)}$$

대칭 키 암호 방식의 경우, 암호화 key K_e와 복호화 key K_d가 동일하다. 그러므로 송신자와 수신자가 비밀 통신을 시작하기 전에 비밀리에 key를 공유하고 있어야 한다. 따라서 사전 비밀리에 key를 공유하기 위한 key 분배 과정이 필요하다. 또한 공개 키 암호 방식은 암호화 key K_e와 복호화 key K_d를 분리

하여 암호화 key K_e 를 공개하고, 복호화 key K_d 를 비밀리에 보관한다. 물론 공개 암호화 key K_e 로부터 비밀 복호화 key K_d 를 계산할 수 없어야 한다. 송신자는 수신자의 공개 암호화 key K_e로 전송하려는 정보를 암호화하고 수신자는 수신한 암호문을 자신의 비밀 복호화 key K_d 로 평문을 복원시킨다.

3. 암호화 도구

3.1 암호화 도구 고찰

암호화 방식은 인류 역사가 시작되기 전부터 사용되어 온 것으로 알려지고 있다. 그러나 국가가 형성되기 전에는 비밀리에 보관해야 할 정보가 그리 많지 않아 암호 방식의 사용이 거의 없었다. 이후 국가가 형성되고, 상업이 발달함에 따라서 개인 및 국가의 이권에 따른 비밀 보전의 필요성이 증대되면서 암호학이 대두되기 시작하였다. 최근에는 컴퓨터의 보급과 정보 통신 기술의 발전으로 정보 시스템을 통한 정보처리, 축적, 전달이 널리 확산됨에 따라 정보 시스템 내에서 정보 보호와 통신 상태의 정보 보호 및 사용자 합법성 확인을 위한 방법으로 암호가 크게 주목을 받고 있다. 특히, 산업 사회로부터 정보화 사회로 전환되면서 개인의 프라이버시, 기업의 경영 비밀 등, 제 3자로부터 보호해야할 정보의 급증으로 정보보호를 위한 기술면에서의 대책인 암호의 사용이 보편화되고 있다. 이들 암호의 변천을 고대 암호와 세계 대전을 전후로 한 근대 암호 및 현대 암호를 나누어 고찰하고자한다.

3.2 고대 암호

현대 암호가 연구되기 전의 고대 암호와 근대 암호는 문자의 전치를 이용한 전치 암호(permutation cipher 또는 transposition cipher)와 문자를 다른 문자로 치환하는 환자 암호(substitution cipher) 그리고, 일정한 방향으로 일정한 간격을 shift 하는 shift cipher 등으로 분류된다. 여기서 가장 오래된 암호가 전치 암호(permutation cipher)로서, 이 방식은 전달하려는 평문(plaintext)을 재배열하는 방식으로 곤봉에 종이를 감아 평문을 횡으로 쓴 다음, 종이를 풀면 평문의 각 문자는 재배치되어 평문의 내용을 인식할 수 없게 된다. 암호문 수신자는 송신자가 사용한 곤봉과 직경이 같은 곤봉에 암호문이 적혀있는 종이를 감고 횡으로 읽으면 평문을 얻을 수 있다. 또한 환자 암호는 로마 시대의 Julius Caesar가 사용한 시이저(Caesar) 암호로서, 평문의 각 문자를 우측으로 3문자씩 이동시켜 그 위치에 대응하는 다른 문자를 치환함으로써 평문을 암호문으로 변환하는 암호 방식이다. 즉, A는 D로, B는 E, C는 F로 계속해서 Z는 C로 평문을 암호문으로 치환하는 방식이다. 이 방식의 암호문에서 평문으로 복호화하는 방법은 암호화의 역 처리, 즉 암호문 문자를 좌측으로 3 문자씩 이동시키면 간단히 평문을 복호화할 수 있다.

3.2.1 Permutation cipher

전치암호(permutation cipher)는 정상적인 평문 배열을 특정한 key 순서에 따라 평문 배열을 재조정하여 암호화하는 방식이다. 먼저 평문 문장을 키의 길이에 따라 일정 간격으로 나눈다. 일정 간격으로 나눈 문자를 키의 재배열 순서에 따라 재배치한다. 만일 일정 간격으로 문자를 나눌 때 마지막 간격의 문자가 모자라면 임의의 문자를 덧붙힌다. 즉 기본 구성으로는, m을 고정된 양의 정수라 정의하고, 평문 (P) $P=C=(Z_{26})^m$ 그리고 K는 $\{0,1,\cdots,m\}$의 모든 permutation으로 구성된다. 또한 key가 π 일 때, 암호문은 $e_\pi(x_1,\cdots\cdots,x_m)=(x_{\pi(1)},\cdots\cdots,x_{\pi(m)})$ 이고, 해독문은 $d_\pi(y_1,\cdots\cdots,y_m)=$

$(y_{\pi^{-1}(1)},\cdots\cdots,x_{\pi^{-1}(m)})$이다. 여기서 π^{-1}는 inverse permutation 이다. 예를들면, m=6이고, 다음의 permutation π 를 key라고 한다.

1	2	3	4	5	6
3	5	1	6	4	2

[그림 2] 정의된 key와 관련 값
[Fig. 2] defined key and connection value

Fig 2에서, 세 번째 문자는 첫 번째, 다섯 번째 문자는 두 번째로, 첫 번째 문자는 세 번째로, 키 순서에 따라 여섯 문자를 재배치한다. 재배치는 평문 전 문장에 대하여 적용되며 재배치된 문장이 암호문이 된다. 암호문을 평문으로 복원하는 복호화 과정은 암호화 과정의 반대 순서로 재배치를 하면 평문이 복원된다. 즉 inversion permutation π^{-1}은 다음과 같다.

1	2	3	4	5	6
3	6	1	5	2	4

[그림 3] 정의된 key와 관련 값
[Fig. 3] defined key and connection value

주어진 평문은 shesellsseashellsbytheseashore 이다. 우선 평문을 6개의 알파벳으로 나누어 그룹을 만든다. 즉, shesel | lsseas | hellsb | ythese | ashore 이다. 이들 각 그룹을 permutation π 에 따라 다시 정리하면, eeslsh | salses | lshble | hsyeet | hraeos 가 된다. 즉 암호문은 eeslshsalseslshblehsyeethraeos 이 된다. 복호화는 같은 방법으로 하며, π^{-1}를 이용한다.

3.2.2 Substitution cipher

환자 암호(substitution cipher)는 평문 문자를 암호문 문자로 일대일 대응시켜 암호화하는 방식으로 평문자를 다른 암호문자로 치환한다. 즉

$P($평문$) = C($암호문$) = Z_{26}, 0 \le K \le 25$일때

각 permutation $\pi \in K$, $e_\pi(x) = \pi(x)$

그리고, $d_\pi(y) = \pi^{-1}(y)$ 여기서 π^{-1}은 inverse permutation 이다. 예를 들어, 임의의 permutation π를 아래와 같이 나타내었다. π는 암호화 함수라고 하며, 평문은 소문자로 그리고 암호문은 대문자로 나타낸 것으로 아래 table 1과 같다.

a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
X	N	Y	A	H	P	O	G	Z	Q	W	B	T	S	F	L	R	C	V	M	U	E	K	J	D	I

<표 1> 평문에 대한 암호문
<Table. 1> ciphertext for plaintext

즉 Table 1에서, $e_\pi(a) = X$, $e_\pi(b) = N$ 이 된다. 그리고 π^{-1}은 복호화 함수이며, 위에서 대문자를 윗줄에 알파벳순으로 나타내며 그에 해당하는 소문자를 아랫줄에 나타낸 것으로 아래 table 2와 같다.

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
d	l	r	y	v	o	h	e	z	x	w	p	t	b	g	f	j	q	n	m	u	s	k	a	c	i

<표 2> 암호문에 대한 평문
<Table. 2> plaintext for ciphertext

즉, $e_\pi(A) = d$, $e_\pi(B) = l$ 과 같이 된다. substitution cipher는 26개의 알파벳 문자의 permutation으로 구성된다.

3.3. 근대 암호

17세기 근대 수학의 발전과 더불어 고급 암호가 발전하기 시작하였으나, 본격적인 근대 수학을 도입한 과학적인 근대 암호는 20세기에 들어와서 발전하기 시작하였다. 불란서 외교관 Vigenere가 고안한 암호 방식, Playfair가 만든

2문자 조합 암호, 오스트리아 육군 대령 Fleissner의 grille 암호 이후, 두 차례의 세계 대전을 거치면서, 암호 방식 설계, 해독에 관한 연구가 활발히 추진되었다. 근대 암호학의 연구를 촉진시킨 것은 두 차례의 세계 대전이 계기가 되었지만, 기술적으로는 전신 기술의 발달과 세계대전 후의 전자계산기의 출현으로 암호화, 복호화 및 암호 해독의 속도가 향상 되므로써 암호 실용화 연구가 활발해졌다. 대표적인 암호에는 Vigenere cipher, Hill cipher 등이 있다.

3.3.1 Vigenere cipher

vigenere cipher는 key의 길이가 m인 알파벳이며, keyword라고 부른다. 따라서 vigenere cipher는 한번에 m개의 알파벳을 암호화한다. 즉, m은 고정된 양의 정수라 놓고, P(평문)$=C$(암 호문)$=K(key)=(Z_{26})^m$로 정의한다. key는 $K=(k_1,k_2,\cdots\cdots,k_m)$라 할 때 암호문 $e_K(x_1,x_2,\cdots\cdots,x_m)=(x_1+k_1,x_2+k_2,\cdots\cdots,x_m+k_m)$이며, 해독문 $d_K(y_1,y_2,\cdots\cdots,y_m)=(y_1-k_1,y_2-k_2,\cdots\cdots,y_m-k_m)$이다. 여기서 모든 연산은 Z_{26}에서 실행된다. 예를 들어, m=6이고, keyword가 cipher라 한다. 즉, K=(2, 8, 15, 7, 4, 17)이다. 평문, thiscryptosystemisnotsecure 일 때, 우선 평문을 정수로 나타내고 key 값을 더하여, modulo 26을 하면 다음의 표와 같이 나타낼 수 있다.

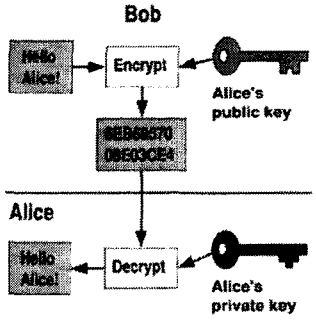
19	7	8	18	2	17	24	15	19	14	18	24	18	19	4	12	8	18	13	14	19	18	4	2	20	17	4
2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15	7	4	17	2	8	15
21	15	23	25	6	8	0	23	8	21	22	15	20	1	19	19	12	9	15	22	8	25	8	19	22	25	19

<표 3> 평문과 키와 암호문
<Table. 3> plaintext and key and ciphertext

따라서 암호화된 암호문은 vpxzgiaxivwpubttmjpwizitwzt이다. 암호문을 다시 복호화 하기 위해서는 같은 keyword를 사용하며, modulo 26의 값을 암호화에서는 더했지만 복호화에서는 빼야 한다. Vigenere cipher에서 길이가 m인 keyword는 26^m 이다.

3.4 현대 암호

현대 암호 방식은 대칭 키(symmetric key)와 비대칭 키(asymmetric key) 암호방식으로 분류된다. 여기서 대칭키 암호 방식은 암호화와 복호화가 한 개의 key로 이루어지고 있으므로 대칭키 암호 방식이라고 한다. 따라서 암호의 안전도는 알고리즘보다는 암호화하는데 사용되는 key의 관리가 중요하다. 여기서 대칭키 암호는 사용되는 키 값과 메시지의 값들을 permutation, substitution 등의 함수를 이용하여 암호화를 수행한다. 가장 잘 알려진 56개의 bits(DES), 3DES, SEED, RC2, RC5 등이 있으며, 최근에 표준화가 완료된 AES(Rijndael) 등이 있다. 비대칭키 암호는 두 개의 키를 사용하는 것으로, 하나는 암호화키로 다른 하나는 복호화 키로 사용한다. 암호화키는 공개키가 되고 복호화 키는 비밀 키가 된다. 공개키 암호에서는 비밀 키 보관에 따라 안전도가 좌우된다. 양 클래스의 장단점을 보면, 대칭키 암호 방식은 처리속도가 매우 빠르다. 그러나 모든 파티(every party)에서 key는 절대적 비밀을 지키게 유지하여야 하며, 이것이 복잡하게 뒤얽여 을 때는 보안에 많은 문제점이 발생한다. 부가적으로, 다른 key를 전달하는 것을 원하는 사람들에게 각 쌍의 key는 동일해야한다. 이것은 key 관리를 매우 부담되게 한다. 이에 반해 비대칭 암호 방식은 조금 더 비용이 많이 드는 값비싼 계산이다, 암호 해독의 key는 모든 사람들에게 분배될 수 있다, 그러나 단지 1명의 사람은 key의 비밀 역할을 억제하여야 한다.



[그림 4] 공개 키와 개별 키 적용
[Fig. 4] public key and private key applied

3.4.1 암호화 기술

암호화 기술은 2개 generic types으로 나눠진다. 즉 대칭 키(symmetric key)와 공개 키(public key)가 있으며, 대칭 키에는 block ciphers와 stream ciphers가 있다. 암호화와 해독 transformations은 $\{E_e : e \in K\}$ 와 $\{D_d : d \in K\}$의 집합으로 구성되어 암호화 계획을 고려한다. 각 K는 key 암호화 영역으로 $A = \{0,1\}$는 유한 집합이다. 각 요소 $e \in K$는 M으로부터 E_e에 의하여 표시된 C까지 유일하게 사상을 결정한다. 여기서 E_e는 암호화 함수, 각 $d \in K$를 위해, D_d는 C로부터 M, 예를 들어, $D_d : C \rightarrow M$까지 사상을 표시한다. 여기서 D_d는 해독 함수이다. 메시지 $m \in M$에 변형 E_e을 적용하는 것의 과정은 암호화 m 또는 m의 암호화하는 것으로서 참조된다. 암호화 계획은 그와 같이 유일한 키 $d \in K$가 있는 각각의 $e \in K$를 위해 속성을 가지고 암호화 변형과 해독 변형의 대응 고정된 $\{D_d : d \in K\}$의 고정된 $\{E_e : e \in K\}$의 그것이 이루어진다. $D_d = E_e^{-1}$ 즉 다시 말하면, 모든 $m \in M$을 위해 $D_d(E_e(m)) = m$이다. 이전 정의에 키들 e와 d는 키 쌍이라고 참조하여, (e, d)으로 표시한다. 구조에 대한 암호화 계획은 하나의 메시지 공간 M, 암호문 공간 C, key 공간 K, 암호화 변형들 $\{E_e : e \in K\}$의 집합, 그리고 해독 변형들 $\{D_d : d \in K\}$의 대응 집합을 선택하도록 요구한다. 암호화 계획은 만약 각자를 위해 연합된 암호/해독 key가 한 쌍으로 되어있다.(e,d) 이것이 대칭 키이다라고 한다. 즉 d를 알고 e를 결정하고, d로부터 e을 결정하는 것은 쉽게 계산된다. 가장 실제적인 대칭성 키 암호화에 e=d의 계획을 세우기 때문에, 대칭 키에 적합하다.

3,5 Bit stream encryption

2진 연산 기호 XOR 연산을 이용하면 문자를 암호화하고 해독하도록 사용할 수 있다. 여기서 암호 작성은, 0 또는 1의 비트에 의해 작성 될 수 있다. 참고로 논리식 T가 1, F가 0으로 정리한다. XOR 연산은 계속된 암호문(ciphertext) 생성을 위하여 XOR 연산을 이용하여 피연산자들로서 평문(plaintext)과 key로부터 계속된 문자들을 이용한다. XOR 암호화(encryption)는 다음과 같이 근본적인 작업에 의해 수행된다. 즉 평문(plaintext)으로부터 문자 즉 단일코드 값들 즉 8비트 2진수 $2^8 = 256$(0~255)로 나타낼 수 있다. 그리고 암호문 작성은 XOR 연산에 비연산자의 평문(plaintext)과 연속 비트를 가지고 있는 key에 의해 제작된다. 예를 들어 만약 평문(plaintext)의 첫 번째 문자가 "c"이면, 2진수는 01100011(10진수는 99)이다. 그리고 key의 첫 번째 문자가 "m"이면, 2진수는 01101101(10진수는 109)이면, XOR 연산을 적용하면, 다음 결과를 얻는다.

비트 수	1	2	3	4	5	6	7	8	결 과
평문 (plaintext) "c"	0	1	1	0	0	0	1	1	$c(99)_{10}$
임의 암호화 key "m"	0	1	1	0	1	1	0	1	$m(109)_{10}$
암호문 (Ciphertext) XOR	0	0	0	0	1	1	1	0	$\hat{}N(14)_{10}$

<표 3> 평문과 키와 암호문
<Table. 3> plaintext and key and ciphertext

암호화 key "m"을 적용하여 평문 "c"를 XOR 연산자를 사용하여 암호한 결과 "00001110"(10진수 14)으로 변화되었다. 이 과정은 각 비트별 XOR 연산자를 이용하여 암호문을 작성하며, 전체 평문이 암호화되었을 때까지 반복한다. 이와 같이 암호문을 작성하여, 암호문의 길이를 줄이기 위하여 2진법 보다 16진법으로 작성한다. 결과적으로 XOR 암호화는 ciphertext-character의 값을 만들기 위해 XORing의 결과에 plaintext-character의 수치 값과 key-character의 수치 값에 결정된다. 또한 Vigenere 암호는 ciphertext-character의 값을 만들기 위해 plaintext-character의 수치 값과 key-character의 수치 값을 더하여, 나머지 255(modulo 255)로 처리한 결과를 결과 값으로 얻는다. 여기서 XOR 연산자가 Vigenere 암호화 기능보다 더 매력적인 것은 더하기 기능을 사용하지 않기 때문이다. 즉 Vigenere 암호 기능은 더하거나 빼는 기능과 나머지를 계산으로 해서 암호화 및 해독하지만, XOR 연산자는 더하거나 빼는 기능 없이 단순히 대칭 연산으로 만 암호화 및 해독한다. 따라서 Vigenere 암호는 비대칭 연산인 반면, XOR 암호화는 대칭 연산을 사용한다. 따라서, XOR을 사용하면 ,쉽게 같은 연산이 문자를 암호화하고 해독할 수 있다. 다음 문장에서, 평문 "p", key "k" 그리고 암호문 "c" 와 OR과 XOR 연산자 사용을 비교하면, 다음과 같이 볼 수 있다. OR 연산자를 사용했을 때, ① p + k = c; ② 그러나 c + k ≠ p; 되며 ③ 그러므로 (p + k) + k ≠ p; ④ 오직 (p + k) - k = p; 이것만이 성립한다. 따라서 OR 연산자를 이용한 암호화 및 해독 사이클은 더하기와 빼기 연산 모두 필요하다. 그러나 XOR 연산자를 사용 했을 때, ① p ⊕ k = c; 되며 ② 그리고 c ⊕ k = p; 되며 ③ 기타 (p ⊕ k) ⊕ k = p; 만족한다. 즉 XOR 연산자를 이용한 암호화 및 해독 사이클은 모든 조건에 만족한다. 다음 예에서 XOR 연산자는 다음 진리표를 보는 것처럼 대칭(symmetric)된 것을 볼 수 있다

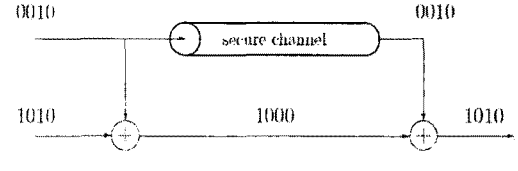
p	k	c
0	0	0
0	1	1
1	0	1
1	1	0

⇒

c	k	p
0	0	0
1	1	0
1	0	1
0	1	1

<표 5> XOR 연산자
<table. 5> XOR operator

다음 그림은 symmetric encryption 방법의 예이다.



[그림 5] XOR 이용한 암호화 과정
[Fig. 5] encryption process that use XOR

암호 작성에 XOR 연산자 사용 예를 보면, 의사 난수 비트 스트림(pseudo-random bit stream) r_i의 XOR을 사용하는 것을 고려하여,

메시지 비트 스트림(message bit stream) m_i을 암호화된 비트 스트림(encrypted bit stream) c_i로 만들며, 조건은 $c_i = r_i \oplus m_i$이다. 또한 해독은 XOR 연산자 이용으로, 의사 난수 비트 스트림(pseudo-random bit stream) r_i와 c_i는 m_i가 된다($m_i = r_i \oplus c_i$). 결국, $r_i \oplus c_i = r_i \oplus r_i \oplus m_i = 0 \oplus m_i = m_i$으로 설명된다.

4. 결론

본 논문은 암호화 원리를 이론적으로 고찰하였다. 암호화 방법에 있어서 과거 암호화의 해독 방법은 단순히 문자를 치환(Permutation cipher 또는 transposition cipher) 하거나 이동하는 하는 방법으로 이용되어 왔으나, 현재는 key stream을 이용하는 방식을 흔히 이용되고 있다. 즉 평문에 key를 생성 적용하여 암호 및 해독한다. 즉 key를 생성하는 방법에 따라서 암호화의 체계가 달라지는 것이다. 따라서 암호화 작성에 새로운 key를 생성 할 수 있는 여러 원리들이 제시될 필요가 있다. 특히 Cellular Automata의 transition rule를 이용하여 다양한 형태의 key를 생성할 수 있다. Cellular Automata는 State, Neighborhood, Program Rules이라는 단순한 성질을 가지고 다양하고 복잡한 원리를 구현할 수 있다. 따라서 본 논문에서는 향후 Cellular Automata를 이용하여 암호화 처리 방법을 다음 논제로 제안하고자 한다.

참고문헌

[1] Beker, Henry and Fred Piper. Cipher Systems, The Protection of Communications. NewYork: Wiley-Interscience Publication, 1982..

[2] J. Seberry and J. Pieprzyk : Cryptography, 1989.

[3] Menezes, A., van Oorschot, P., Vanstone, S. *Handbook of Applied Cryptography*, CRC Press, 1997.

[4] R. L. Rivest, A. Shamir and L. Adleman, "A method of obtaining digital signature and public key cryptosystem", ACM Communication 21 No.2, pp.120-126, 1978.

[5] "Vigenère Cipher". Wikipedia: The Free Encyclopedia. 21 Feb 2005, 21:58 UTC. 24 Feb 2005.
<http://en.wikipedia.org/wiki/Vigenere_cipher>

남 태 희

1993~현재: 동주대학 의료기공학과 부교수
1996~ 부경대학교 전자공학과 박사수료

관심분야 : Cryptography, Cellular Automata, 데이터베이스, 전자상거래, 패턴인식, MIS, GIS, ERP, 보건의료정보학