제 2 장 고전암호알고리즘

메시지의 비밀성을 보장하기 위해 암호화하는 기술은 고대 로마시대부터 사용되었다. 주로 암호기술은 군사적 목적으로 많이 사용되었다. 이 처럼 2차 세계 대전 이전에 사용된 암호기술들을 고전 암호기술이라 한다. 고전 암호기술 중 암호알고리즘들은 주로 **치환**(substitution) 암호방식이거나 **자리바꿈**(permutation) 암호방식을 사용하였다. 치환 암호방식은 평문의 정보를 다른 정보로 바꾸어 암호화하는 방식이고, 자리바꿈 암호방식은 평문을 구성하는 정보들의 위치를 바꾸어 암호화하는 방식이다.

2.1 치환암호방식

가장 기초적인 치환 암호방식은 **단순치환**(monoalphabetic) 암호방식이다. 고전 암호알고리즘에서 평문은 주로 영문 데이터이며, 수작업으로 보통 암호화가 이루어졌다. 단순치환 암호방식은 영문의 각 문자를 다른 문자로 바꾸어 암호화하는 방식이다. 가장 대표적인 단순치환 암호알고리즘은 이동(shift) 암호알고리즘이다. 이 알고리즘은 평문의 각 영문자를 암호키에 해당되는 수만큼 이동하여 암호화한다. 이와 같은 이동 암호방식은 고대 로마시대부터 사용되었으며, 줄리우스 카이사르(Julius Caesar)는 키를 3으로 사용하였다고 하여, 키가 3인 경우를 카이사르 암호방식이라 한다. 하지만 키는 0-25까지 임의의 수를 사용할 수 있다. 키가 3인 경우에 'A' 문자는 'D'로, 'B' 문자는 'E'로 바꾸어 표현함으로써 암호화된다. 물론 키 값으로 0을 사용하면 평문이 전혀 바뀌지 않으므로 의미가 없다. 따라서 이동 암호알고리즘은 총 25개의 키가 존재한다. 하지만 실제 단순치환 암호알고리즘은 총 26!개의 키가 존재한다. 하지만 이동 암호방식처럼 어떤 규칙이 없으면 그 키를 암기하기가 어렵다는 문제점이 있다. 이처럼 키가 복잡해지면 안전성은 증가하지만 키를 안전하게 유지하는 것이 어려워질 수 있다.

임의의 맵을 사용하는 단순치환 암호알고리즘은 총 26!개의 키가 존재하므로 전사공격에 대해 안전하다고 생각할 수 있다. 하지만 실제 안전하지 않다. 단순치환 암호방식의 가장 큰 문제점은 평문의 문자 출현 빈도를 숨길 수 없다는 것이다. 즉, 단순치환 암호방식에서 특정 문자는 늘 고정된 다른 문자로 암호화된다. 따라서 'A'문자가 'X'로 암호화된다고 가정하였을 때 평문에 'A'문자가 10번 등장하면 암호문에도 'X'문자가 10번 등장한다. 이와 같은 정보는 암호해독에 매우 유용한 정보가 된다. 따라서 키의 개수가 많아져 모든 경우의 수를 검사하여 키를 알아내는 것이 어렵다고 하여 무조건 암호방식이 안전한 것은 아니다.

이 문제를 극복하기 위해 몇 가지 방법이 제시되었다. 그 중 동음(homophonic) 치환 암호방식은 한 문자를 여러 개의 다른 값으로 매핑하여 암호화하며, 다중문자(polygram) 치환 암호방식은 문자들을 블록단위로 매핑한다. 한다. 동음치환 방식의 경우에는 평문의 가능한 문자보다 암호문의 가능한 문자의 수가 커진다. 동음 치환 암호방식이나 다중문자 치환 암호방식은 단순치환 암호방식의 출현 빈도 노출 문제를 극복할 수 있지만 암호키의 표현이 어려우지는 단점을 지니고 있다.

단순치환 암호방식의 문제점을 근본적으로 해결하기 위한 가장 효과적인 방법은 **다중치환**(polyalphabetic) 암호방식이다. 이 방식에서는 주기적으로 문자의 위치마다 다른 이동 암호방식을 사용한다. 다중치환 암호방식의 가장 대표적인 알고리즘은 Vigenere 암호방식이다.

이 암호방식에서는 키 길이가 5이면 매 다섯 문자마다 같은 키로 이동 치환하여 암호화한다. 예를 들어 K=(2,8,15,7,4,17)이고 평문이 “thiscryptosystemisnotsecure"이면 결과 암호문은 ”VPXZGIAXIWPUBTTMJPWIZITWZT"이다. 즉, 첫 문자인 't'와 일곱 번째 문자인 ‘y' 등은 모두 키 2로 암호화된다,

2.2 자리바꿈 암호방식

자리바꿈 암호방식은 평문을 구성하는 요소들의 위치를 바꾸는 암호방식을 말한다. 가장 단순한 자리바꿈 암호방식은 다음과 같은 열기준 암호방식이 있다.

- **단계 1.** 평문의 작업 단위를 결정한다. 예) 여섯 문자
- **단계 2.** 평문을 정해진 열 단위로 표현한다.

  예2.1) LASTNITEWASHEAVENPLEASEMARRYME

L	A	S	T	N	I
T	E	W	A	S	H
E	A	V	E	N	P
L	E	A	S	E	M
A	R	R	Y	M	E

- **단계 3.** 열 기준으로 다시 표현한다.

  예2.2) LTELA AEAER SWVAR TAESY NSNEM IHPME

자리바꿈 암호방식도 추측 공격이 가능하다. 자리바꿈 암호방식의 안전성을 높이기 위해서는 자리바꿈을 여러 번 반복할 수 있다. 위 예에서 동일한 방식으로 한번 더 자리바꿈을 하면 다음과 같다.

예2.3) LTELA AEAER SWVAR TAESY NSNEM IHPME

L	T	E	L	A	A
E	A	E	R	S	W
V	A	R	T	A	E
S	Y	N	S	N	E
M	I	H	P	M	E

→ LEVSM TAAYI EERNH LRTSP ASANM AWEEE

컴퓨터에서 자리바꿈 암호는 보통 자리바꿈 맵을 사용한다. 자리바꿈 맵을 사용할 경우에도 작업 단위를 먼저 결정해야 한다. 단순 자리바꿈처럼 작업 단위를 여섯 문자라고 가정하면 키는 다음과 같이 표현할 수 있다.

예2.4)

x	1	2	3	4	5	6
π(x)	2	4	5	1	6	3

평문: LASTNI TEWASH EAVENP LEASEM ARRYME

암호문: ATNLIS EASTHW AENEPV ESELMA RYMAER

치환 암호방식과 자리바꿈 암호방식을 결합하여 사용하면 안전성을 높일 수 있다. 이 처럼 치환과 자리바꿈을 함께 사용하는 암호방식을 **혼합**(product) 암호방식이라 하고, 현대 대칭 암호알고리즘은 모두 혼합 암호방식을 사용하고 있다.

2.3 고전 암호알고리즘에 대한 암호해독

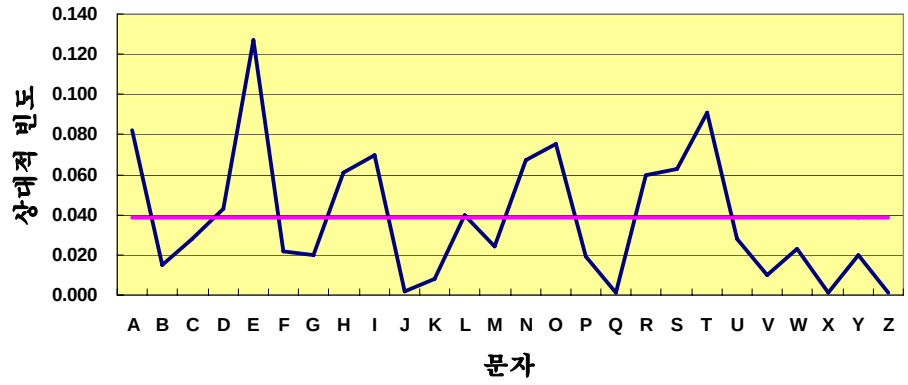
이 절에서는 다중치환 암호방식으로 암호화된 암호문을 해독하는 방법을 살펴봄으로써 기본적인 암호해독 기술을 경험해본다. 다중치환 암호방식으로 암호화하였다는 것은 암호문에 대응되는 평문은 일반 영문이라는 것을 의미한다. 암호해독을 할 때 그것의 평문 내용을 알 수 없지만 평문의 형태를 알고 있다면 암호해독에 많은 도움을 줄 수 있다. 특히, 평문의 특징이 암호문에 그대로 또는 일부 나타날 수 있으므로 이것을 암호해독할 때 활용해야 한다.

영문의 가장 큰 특징은 등장하는 각 문자의 출현 빈도가 일정하다는 것이며, 다음과 같은 특징을 가지고 있다.

- ‘e’, ‘t’, ‘a’, ‘o’ 등은 다른 문자들에 비해 출현 빈도가 매우 높다.
- 단어의 시작에서 가장 많이 사용되는 문자는 ‘t’이고, 단어 끝에 가장 많이 사용되는 문자는 ‘e’이다.
- 길이가 1인 단어는 ‘a’와 ‘I’가 유일하다.
- 길이가 2인 단어 중 가장 많이 사용되는 것은 “to”와 “in”이다.
- 세 개의 문자 조합 중 가장 많이 사용되는 것은 “ing”, “ion”, “ent”이다.
- 모음 다음에 가장 많이 등장하는 문자는 ‘n’이다.

<표 2.2> 영문에서 각 문자의 등장 빈도

문자	확률	문자	확률
A	.082	N	.067
B	.015	O	.075
C	.028	P	.019
D	.043	Q	.001
E	.127	R	.060
F	.022	S	.063
G	.020	T	.091
H	.061	U	.028
I	.070	V	.010
J	.002	W	.023
K	.008	X	.001
L	.040	Y	.020
M	.024	Z	.001



<그림 2.1> 일반 영문에서 각 문자의 등장 빈도에 따른 분포

표 2.1은 일반 영문에서 각 문자가 등장하는 비율이며, 그림 2.1은 각 문자 등장 빈도에 따른 분포를 나타내고 있다. 따라서 일반 영문 데이터를 이동 암호방식을 이용하여 암호화하면 결과 암호문에서 각 문자의 등장 빈도에 따른 분포를 그리면 그림 2.1의 분포도를 일정하게 이동시킨 형태가 된다. 따라서 특별한 어려움 없이 키를 알아낼 수 있다.

다중치환 암호방식으로 암호화된 암호문을 해독하는 방법은 크게 두 가지 원리를 이용한다. 첫째, 다중치환은 단순치환의 반복이다. 둘째, 단순치환은 영문의 등장 빈도를 숨길 수 없으므로 앞서 언급한 바와 같이 영문의 특성을 이용하여 쉽게 해독할 수 있다. 따라서 다중치환 암호방식으로 암호화된 암호문을 단순치환 암호방식으로 암호화된 여러 개의 암호문으로 나눌 수만 있다면 이들을 개별적으로 해독함으로써 전체를 해독할 수 있다. 다중치환 암호방식으로 암호화된 암호문을 단순치환 암호방식으로 암호화된 암호문으로 나누기 위해서는 사용된 키 길이를 알아내야 한다. 다중치환 암호방식의 암호문에서 키 길이는 두 가지 방법으로 예측이 가능하다. 하나는 Kasiski 방법이고, 다른 하나는 IC(Index of Coincidence) 방법이다.

Kasiski 방법은 영문에서 높은 빈도로 등장하는 패턴을 이용하는 방법이다. 기본 원리는 길이가 n인 키를 사용하여 메시지를 다중치환 암호방식으로 암호화할 때, 어떤 특정 단어 또는 문자 그룹이 평문에 k번 등장하면 그것은 같은 문자로 거의 k/n번 암호화된다는 것이다. 예를 들어 “the"라는 단어 또는 문자그룹이 평문에 n번 이상 등장하면 최소 두 번은 동일한 키로 암호화된다는 것이다. 따라서 암호문에서 반복하여 나타나는 길이가 3이상인 영문 패턴을 찾아 그 위치를 기록하고, 나타난 위치 간에 공약수를 찾아 키의 길이를 결정할 수 있다.

예2.5) 암호문이 다음과 같다고 가정하자

```
vyclh tufpv mvpwt ujckx trjsx hzlxm kflhb vlqtw vfztt efkee
kdccm vyccb vscrt ovycb pjsam nrrte agcdi nvqtx okmab mvagt
ebcgy qirwx drbvn ajycw jrazx twmgm jvedh fxsnl kucub pvywt
ebcgt urlxg fztxw wrjla qvvex tzktg vjuxm jkfte kdgit vzmcl
qwqnl vvkhy qigcm gcjtv vlyav wigdl kkwdk uyctk rccpl wicia
gnmgw fvqrk kscht rvphh pngia cgygm ktsat tjcih hjixe njycw
pfrpi cirxv wcygl gkmuf qiyal vycgx cicvh qufpv mvpht puzpw
```

jrazx tjhjl vrqia gicpk gxmdw rcsbx tjycw drbee wdcgl

“vyc”와 “jycw"가 암호문에 등장한 횟수, 위치, 위치 간의 차이는 다음과 같다.

vyc		
시작 위치	차이	약수
1		
56	55	1,5,11
67	11	1,11
326	259	1,7,37

jycw		
시작 위치	차이	약수
117		
297	180	1,2,3,5
382	85	1,5,17

위 표에서 “vyc"의 경우에는 공통약수가 1이지만 키 길이가 1이 아닐 확률이 많기 때문에 ”vyc" 중에 대응되는 평문이 다른 경우가 존재한다는 것을 의미한다. 반면에 “jycw"는 공통약수에 5가 있으므로 키 길이가 5라고 예측할 수 있다.

IC 방법은 단순치환 암호방식으로 암호화된 암호문에 등장하는 문자 빈도는 이것에 대응되는 평문의 문자 빈도와 같다는 원리를 이용한 방법이다. 즉, 일반 영문에서 각 문자의 등장 빈도에 따른 분포는 그림 2.1과 같다. 이것을 이동 암호방식을 이용한 단순치환 암호방식으로 암호화하여 암호문의 문자 빈도 분포를 그리면 그림 2.1의 분포를 왼쪽 또는 오른쪽으로 이동된 형태가 된다. 다중치환 암호방식에서 키 길이가 길어지면 길어질수록 평문에 등장하는 문자 빈도 분포와 암호문에 등장하는 문자 빈도 분포가 다르게 된다. 특히, 길이가 길어지면 길어질수록 암호문의 등장하는 문자 빈도 분포는 균등 분포가 된다. 따라서 암호문의 등장하는 문자 빈도 분포를 조사하여 키 길이가 어느 정도인지 예측할 수 있으며, 이것을 IC 방법이라 한다.

Prob_a가 문자 ‘a'가 평문에 등장할 확률이면 $\Sigma_{\lambda=a}^{\lambda=z}\text{Prob}_\lambda = 1$이다. 모든 영문자가 동일한 확률로 등장하면 각 문자가 평문에 등장할 확률은 모두 0.0384가 된다. 현재 분포가 균등분포와 얼마나 차이가 나는지 평가하기 위해서는 표준편차를 이용해야 한다. 즉, 다음 식을 이용하여 표준편차를 계산해야 한다.

$$\text{var} = \sum_{\lambda=\text{a}}^{\lambda=z}\left(\text{Prob}_\lambda - \frac{1}{26}\right)^2 = \sum_{\lambda=a}^{\lambda=z}\text{Pro}b_\lambda^2 - \frac{1}{26}$$

이 값을 계산하기 위해서는 Prob_λ^2를 계산해야 하며, 이 값은 $\frac{\text{Freq}_\lambda(\text{Freq}_\lambda-1)}{n(n-1)}$으로 추정할 수 있다. 여기서 $\text{Freq}_{\lambda=a}$는 문자 'a'가 암호문에 등장한 빈도이고, n은 암호문의 길이이다. 이 때 IC는 다음과 같이 정의되며,

$$\text{IC} = \sum_{\lambda=a}^{\lambda=z}\frac{\text{Freq}_\lambda(\text{Freq}_\lambda-1)}{n(n-1)}$$

키 길이와 IC와의 관계는 표 2.2와 같다.

<표 2.2> IC와 키 길이의 관계

키의 길이	1	2	3	4	5	10	>10
IC	.068	.052	.047	.044	.044	.041	.038

예2.6) 예2.5의 암호문에서 각 문자의 등장 빈도를 계산한 후에 IC값을 계산하면 IC = 0.440275이다. 따라서 Kasiski 방법을 통해 예측한 키 길이가 올바르다는 것을 알 수 있다.

Kasiski 방법과 IC 방법을 이용하여 다중치환 암호방식으로 암호화된 암호문을 해독하는 방법은 다음과 같다.

- **단계 1.** Kasiski 방법을 이용하여 키 길이를 예측한다.
- **단계 2.** 예측한 키 길이의 올바름을 IC를 구하여 확인한다.
- **단계 3.** 암호문을 해독하기 위해 암호문을 키 길이만큼 나눈다.
- **단계 4.** 나누어진 각 부분은 이동 암호방식으로 암호화되어 있다. 따라서 각 부분에서 각 문자의 등장 빈도를 계산한 다음에 가장 많이 등장한 문자를 e, t, a, o, n 순으로 매핑하면서 키를 구한다.

2.4 XOR 연산을 이용한 암호알고리즘

XOR 연산은 $A \oplus A = 0$, $A \oplus 0 = A$라는 특성을 가지고 있다. 따라서 $A \oplus B \oplus B = A$이다. 이 원리를 이용하여 다음과 같은 알고리즘을 통해 메시지를 암호화하고 복호화할 수 있다.

- 암호화 알고리즘: $E_K(P) = P \oplus K$
- 복호화 알고리즘: $D_K(C) = C \oplus K$

만약 키 길이가 짧고 일반 영문을 이와 같은 방법으로 암호화할 경우에는 다중치환 암호방식처럼 쉽게 해독할 수 있다. 이 해독 방법에서도 가장 중요한 것은 키 길이를 구하는 것이다. 키 길이는 다중치환 암호방식처럼 Kasiski나 IC 방법을 사용할 수 있다. 키 길이가 결정되면 암호문과 키의 길이만큼 이동한 암호문을 XOR한다. 이렇게 하면 평문 간에 XOR한 문자열을 얻을 수 있으며, 평문 간에 XOR한 문자열은 쉽게 해독할 수 있다. 그러나 키 길이가 매우 길면 키 길이를 예측하기 어려워 이와 같은 해독 공격을 하기 어렵다. XOR를 이용한 암호방식에서 동일한 키를 주기적으로 반복하여 암호화하지 않고 매번 평문과 같은 길이의 다른 키를 사용하여 암호화하는 방식을 one-time pad 암호방식이라 한다. 이 암호방식은 암호문 단독 공격에 대해서는 무조건적으로 안전하다는 것이 증명되어 있다. 하지만 무한하고 랜덤한 키를 사용하기 때문에 이것을 공유하기가 어렵다는 단점을 가지고 있다.

2.5 혼돈과 확산

Shannon은 암호알고리즘 설계의 두 가지 기본 원칙인 혼돈(confusion)과 확산(diffusion) 이론을 제시하였다. 혼돈이란 평문과 암호문과의 상관관계를 숨기는 것을 말하며, 이를 통

해 암호문과 암호키 간에 관계를 알기 어렵게 만든다. 확산이란 평문의 통계적 특성을 암호문 전반에 확산시켜 이를 숨기는 것을 말하며, 이를 통해 암호문과 평문 사이의 관계를 어렵게 만든다. 단순치환은 혼돈 특성만 갖으며, 자리바꿈은 확산 특성만 갖는다. 따라서 Shannon은 혼돈과 확산 특성을 동시에 갖도록 치환과 자리바꿈을 결합하여 사용해야 한다는 것을 제시하였으며, 안전성을 높이기 위해 연산들을 반복적으로 사용할 것을 제시하였다. 이 이론은 오늘날까지 계속 적용되고 있다.

	1	2	3	4	5
1	a	b	c	d	e
2	f	g	h	i/j	k
3	l	m	n	o	p
4	q	r	s	t	u
5	v	w	x	y	z

<그림 2.2> Polybious Square

Polybious는 Polybious square를 이용하는 단순치환 암호방식을 제안하였다. 예를 들어 그림 2.2와 같은 Polybious square를 사용한다고 가정하자. 그러면 'a' 문자는 11로 'b‘ 문자는 23 등 문자를 그것의 행과 열 번호로 매핑하여 암호화한다. 따라서 “hello"를 그림 2.2의 square를 이용하여 암호화하면 ”23 15 31 31 34“가 된다. 결과 암호문을 보면 이것의 대응되는 평문은 ”sunny", hobby"와 같이 셋 번째와 네 번째 문자가 같은 단어를 암호화한 것이라고 알 수 있다. 이 처럼 치환은 평문의 통계적 특성이 그대로 나타난다. 만약 결과 암호문에 추가적으로 자리바꿈 맵 (5, 3, 4, 1, 2)을 이용하여 한 번 더 암호화하면 “34 31 31 23 15“가 된다. 즉, 치환하였을 때 나타난 평문의 특성이 이제는 사라졌다.

참고문헌

[1] C. Shannon, "Communication Theory of Secrecy Systems," Bell Systems Journal, Vol. 28. pp. 659-715, 1949.