# 2장

# 암호 기초

MXDXBVTZWVMXNSPBQXLIMSCCSGXSCJXBOVQXCJZMO
JZCVCTVWJCZAAXZBCSSCJXBQCJZCOJZCNSPOXBXSB
TVWJCJZDXGXXMOZQMSCSCJXBOVQXCJZMOJZCNSPJZ
HGXXMOSPLHJZDXZAAXZBXHCSCJXTCSGXSCJXBOVQX

*Lewis Carroll의 평문, 이상한 나라의 앨리스*

이 글자들은 한번 척 보는 것만으로 해답을 알아내는 것은 불가능하다.
보는 바와 같이 소위 말하는 암호문인데, 분명 어떤 의미를 내포하고 있다.

*에드가 앨런 포우, 황금벌레*

## 2.1 개요

이 장에서는 암호에 대한 몇 가지 기본적인 요소를 다룬다. 여기서 다져진 암호 관련 기초 지식은 이 책의 다른 암호 관련 내용을 이해하기 위한 토대가 될 것이다. 어려운 수학적 설명은 가능한 생략하겠지만 정보보안이 "무엇"인가 뿐만 아니라 "왜" 그렇게 해야 하는가를 이해하는 데 필요한 세부 내용은 충분히 제공할 것이다.

이 장에서 암호에 대한 개요를 다루는데, 암호에 구체적으로 설명하는 장에서는 대칭키 암호, 공개키 암호, 해시함수, 고급 암호분석 등 4가지가 주제가 될 것이다.

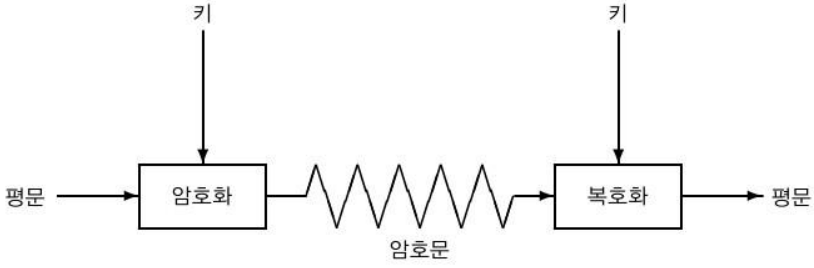
## 2.2 암호 용어

암호를 다룰 때 사용되는 기본 용어는 다음과 같다.

- 암호학(Cryptology): "비밀코드"를 만들거나 해독하는 학문

- **암호생산**(Cryptography): "비밀코드"를 만드는 것[1]
- **암호분석**(Cryptanalysis): "비밀코드"를 해독하는 것
- **암호**(Crypto): 위 세 가지 모두 또는 그 이상을 의미하거나 그 중 특정 용어의 유사어로 사용된다. 따라서 정확한 의미는 사용되는 문맥으로 파악할 수 있다.

**암호**나 **암호체계**는 데이터를 **암호화**하는 데 사용된다. 원본 데이터는 **평문**이며 암호화된 데이터는 **암호문**으로 정의된다. 암호문을 원래의 평문으로 복원하는 것을 **복호화**라고 한다. **키**는 암호화와 복호화를 위한 암호체계를 구성하는 데 사용된다. **대칭키 암호체계**는 [그림 2-1][2]의 블랙박스 암호체계와 같이 암호화와 복호화에 같은 키를 사용하는 기법을 말한다.




**[그림 2-1] 암호체계의 블랙박스**

암호화와 복호화에 각각 다른 키를 사용하는 **공개키 암호**체계라는 개념도 있다. 복호화에 다른 키를 사용하기 때문에 암호화에 사용되는 키를 공개할 수 있다. 공개키 암호체계에서 암호화를 위한 키를 **공개키**라고 한다. 반면, 비밀이 유지되어야 하는 복호화 키는 **개인키**라는 용어를 사용한다. 대칭키 암호체계에서 키는 **대칭키**로 정의하고자 한다. 이렇게 함으로써 **비밀키**라는 용어가 가지는 모호성을 배제할 수 있다.

모든 암호체계의 목표는 키 외에는 복호화할 수 있는 방법이 없도록 하는 것이다. 다시 말해 공격자인 트루디가 사용된 암호 알고리즘을 완전히 파악하고 있고 이와 관련된 정보를(추후에 상세히 설명된다) 많이 가지고 있더라도 키 없이는 복호화해 평문을 얻을 수 없어야 한다. 현실에서는 이와는 다른 상황이 가끔 발생하기도 하지만, 원칙적으로 그렇게 하는 것이 암호체계의 목표이다.

[1] **역자주**: cryptography는 암호를 만드는 것을 의미하는 단어로 여기서는 "암호생산"이라고 표현했지만 일반적으로는 "암호학"과 같은 의미로 사용된다.

[2] 이것이 이 책에서 유일하게 포함된 블랙박스이다.

암호에서의 근본 원칙은 공격자인 트루디가 키만 빼고 암호체계 내부에서 작동되는 모든 것을 완전히 알고 있다고 가정하는 것이다. 이것을 **커크호프 원칙**(Kerckhoffs Principle)이라고 하는데, 그 명칭은 참고문헌 [128]번에서 암호체계 설계 및 사용의 6가지 원칙을 제시한 저자의 이름에서 따온 것이다. 커크홉스라는 이름을 유명하게 만든 이 원칙은 "암호체계 자체가 비밀이 되어서는 안된다. 이것은 적의 손에 어렵지 않게 들어갈 수 있기 때문이다"라는 주장에 근거하고 있다[127]. 즉, 암호체계의 설계 자체가 비밀이 되어서는 안된다는 것이다.

커크호프 원칙이 의미하는 것은 무엇일까? 트루디가 암호체계의 작동 방법을 모르면 피곤할 수 밖에 없다. 또한 암호체계의 세부 내용이 오랜 세월 동안 밝혀지지 않은 사례가 있는 것도 사실이다. 그러나 역공학을 통해 소프트웨어의 알고리즘을 쉽게 찾아낼 수 있고, 함부로 손댈 수 없도록 하드웨어에 내장된 알고리즘도 이와 유사한 방법으로 쉽게 노출될 수 있다. 아울러 지적할 점은 비밀로 다루어진 암호 알고리즘이 일단 공개되고 나면 그 안전성이 실패한 사례가 많다는 사실이다. 참고문헌 [26]번을 보면 이러한 사례를 많이 찾아볼 수 있다. 이러한 이유로 인해, 암호 분야에서는 어떤 암호 알고리즘이 많은 암호학자에 의해 장기간 세부적으로 수행된 분석에도 잘 견디어 낼 때까지는 그 알고리즘을 안전하다고 인정하지 않는다. 결론적으로, 어떠한 암호체계도 커크호프 원칙을 충족하지 않는다면 결함이 있는 것으로 인정된다. 즉, 암호체계는 "무죄가 증명될 때까지는 유죄"이다.

커크호프 원칙을 암호뿐만 아니라 다른 보안 분야에도 확대해 적용할 수 있다. 즉, 커크호프 원칙은 보안 설계 자체는 공개되어 있다는 것을 의미한다. 더 많은 사람이 분석하면 할수록 보안상 결점을 더 확실하게 발견할 수 있다는 점은 명확하다. 커크호프 원칙이 일반적인 원칙으로 받아들여지고 있지만, 실제로는 이러한 근본 원칙을 위배하고자 하는 유혹이 많다. 이 책에서 이에 대한 여러 가지 실례를 살펴볼 것이다.

다음 절에서는 몇 가지 고전 암호체계를 간단히 살펴본다. 암호의 역사 자체만으로도 매우 매혹적인 주제[122]이긴 하지만, 여기서 소개되는 내용은 현대 암호에 담겨 있는 몇 가지 중요한 개념에 대한 기초를 제공할 것이다.

## 2.3 고전 암호

이 절에서는 각기 다른 특징을 가지고 있는 고전 암호체계 4가지를 살펴볼 것이다. 첫 번째는 단순치환(Simple Substitution)인데, 이것은 적어도 2000년 이상된 가장 오래된 암호체계 가운데 하나로서 기본 공격 방법을 설명하기에도 적절하다. 그 다음에는 현대 암호에서 사용되는 중요한 개념을 담고 있는 이중전위(Double Transposition) 암호로 관심을 돌릴 것이다. 이어서 고전 암호인 코드북을 설명할 것이다. 많은

현대 암호는 거대한 전자 코드북과 같은 개념으로 생각될 수 있기 때문이다. 마지막으로 실제 암호체계 중에서 유일하게 이론적으로 안전성이 증명될 수 있는 일회성 암호를 살펴보고자 한다.

### 2.3.1 단순치환 암호

간단한 치환 암호에 대한 전형적인 방법은 현재 문자를 $n$번째 앞에 있는 문자와 서로 치환해 암호화하는 것이다. 예를 들어, $n = 3$을 키로 사용하면 치환되는 문자는 다음과 같다.

```
평문:  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
암호문: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

관례에 따라 평문은 소문자로, 암호문은 대문자로 표기하기로 한다. 위 예에서는 키는 문자의 이동 자리 수를 나타내는 "3"이라고 할 수 있다.

다음과 같은 평문이 있다고 가정하자.

```
fourscoreandsevenyearsago
```

$n = 3$을 키로 이용하면, 평문 열에 있는 각 문자를 그 문자의 3번째 앞에 있는 문자로 치환한다. 이렇게 만들어진 암호문은 다음과 같다.

```
IRXUVFRUHDAGVHYHABHDUVDIR
```

여기서 이 암호체계가 왜 단순치환 암호로 불리는지 분명히 이해할 수 있을 것이다.

복호화는 암호문의 암호화된 문자 각각을 알파벳상으로 3번째 뒤에 위치한 문자로만 치환하면 된다. $n = 3$을 이용한 단순치환 암호는 줄리우스 카이사르가 즐겨 사용했기 때문에 카이사르 암호로도 알려져 있다.

만약 단순치환을 문자를 이동시키는 방법에만 한정한다면, 가능한 키는 $n \in \{0,1,2,...,25\}$이다. 트루디가 다음과 같은 암호문을 도청하였다고 가정하자.

```
CSYEVIXIVQMREXIH
```

이 암호문이 "$n$ 이동"의 단순치환 암호문이라 추측했다고 하자. 그러면 트루디는 추정 키 각각에 대해 그 문장을 복호화해 평문화된 문장이 일반적으로 인식될 수 있는 문장인지를 점검하는 방법으로 26개의 가능한 키 모두를 사용해 볼 수 있다. 그 문장이 $n$ 이동 방법으로 암호화되어 있다면, 평균 13번 시도로 그 키를 찾아낼 수 있다.

올바로 해독된 평문을 우연히 찾을 때까지 가능한 키를 모두 시도해 보는 방법을 전수키조사라고 한다. 그런데 이러한 공격 방법은 언제든지 시도할 수 있기 때문에, 트루디가 일정 시간 이내에 공격을 시도하기 곤란할 정도로 많은 수의 키가 필요하다.

그렇다면 키 공간이 얼마나 커야 충분하다고 할 수 있을까? 트루디에게 1초에 $2^{40}$개의 키를 검사할 수 있는 아주 빠른 컴퓨터가 있다고 가정하자.[3] 그러면 키 공간이 $2^{56}$이면 $2^{16}$초, 즉 18시간 정도 소요되고 $2^{64}$이면 반 년 이상이 소모될 것이다.

단순치환 암호를 $n$ 이동 방법으로 한정할 필요는 없다. 26개 문자의 순열도 키로 사용할 수 있다. 예를 들어 아래의 키는 문자를 이동한 것이 아니면서 단순치환 암호 범주 안에 있다.

```
평문:  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
암호문: Z P B Y J R G K F L X Q N W V D H M S U T O I A E C.
```

만약 단순치환 암호가 모든 순열을 하나의 키로 사용할 수 있다면 가능한 키의 수는 $26! \approx 2^{88}$개이다. 초당 $2^{40}$개의 키를 테스트할 수 있는 초고속 컴퓨터를 사용해도 $2^{88}$개의 키 공간은 890만년이 더 걸린다. 물론 확률적으로 바른 키를 찾는 데는 그 시간의 절반이면 되지만 그래도 495만년이 걸린다. $2^{88}$개는 트루디가 적정한 시간 내에 점검할 수 있는 양과는 거리가 멀기 때문에 그 키 공간은 전수키조사를 불가능하게 할 정도로 충분히 크다.

그러면 이것이 단순치환 암호가 안전함을 의미하는 것일까? 대답은 "아니다"이다. 다음에 소개하는 단순치환 암호체계에 대한 공격이 그 대답이 될 것이다.

---

[3] 1998년에 전자 프런티어 재단(EFF, Electronic Frontier Foundation)이 DES(Data Encryption Standard, 다음 장에서 설명한)를 해독하기 위해 특수 목적의 키 해독 기계를 만들었다. 이 기계는 가격이 $220,000이고, 43,200개의 CPU로 구성되며, 각각이 40MHz 속도로 초당 2백5십 만개의 키를 점검할 수 있다[119]. 하나의 CPU가 4GHz인 현대적인 PC를 생각할 때 이것을 추론해 보면 트루디는 그러한 기계 한 대에서 초당 키를 $2^{30}$개보다 적게 점검하는 셈이다.

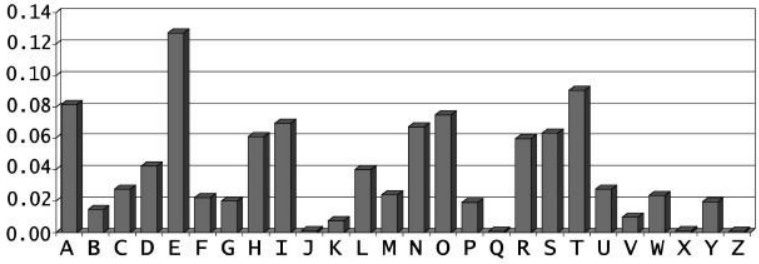
### 2.3.2 단순치환에 대한 암호분석

트루디가 다음의 단순치환 암호문으로 추정되는 문장을 도청했다고 가정하자. 이 암호문이 반드시 $n$ 자리수 이동에 의해 생성될 필요는 없다.

```
PBFPVYFBQXZTYFPBFEQJHDXXQVAPTPQJKTOYQWIPBVWLXTOXBTFX
QWAXBVCXQWAXFQJVWLEQNTOZQGGQLFXQWAKVWLXQWAEBIPBFXFQV
XGTVJVWLBTPQWAEBFPBFHCVLXBQUFEVWLXGDPEQVPQGVPPBFTIXP
FHXZHVFAGFOTHFEFBQUFTDHZBQPOTHXTYFTODXQHFTDPTOGHFQPB
QWAQJJTODXQHFOQPWTBDHHIXQVAPBFZQHCFWPFHPBFIPBQWKFABV
YYDZBOTHPBQPQJTQOTOGHFQAPBFEQJHDXXQVAVXEBQPEFZBVFOJI
WFFACFCCFHQWAUVWFLQHGFXVAFXQHFUFHILTTAVWAFFAWTEVOITD
HFHFQAITIXPFHXAFQHEFZQWGFLVWPTOFFA
```

(2-1)

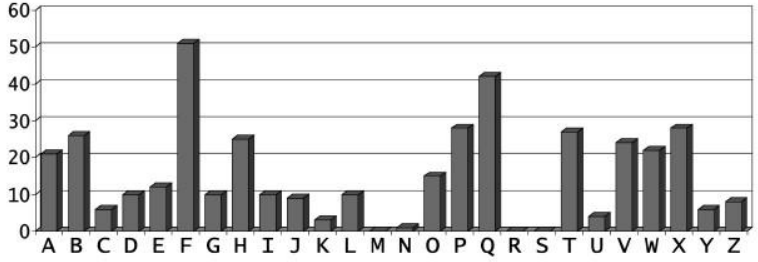
트루디가 $2^{88}$번 시도하는 것은 실질적으로 불가능하기 때문에 다른 현명한 방법을 찾아야 할 것이다. 주어진 문장이 영어로 쓰여졌다고 가정하자. 트루디는 [그림 2-2]와 같은 영어문자 빈도수와 함께 [그림 2-3]과 같은 암호문 (2-1)의 빈도수를 찾아낼 수 있다.




**[그림 2-2] 평문 영어 문자 빈도수 그래프**

암호문의 빈도수로부터, 트루디는 암호문 중에서 "F"문자가 가장 많이 쓰이는 문자임을 알 수 있고, 한편으로 [그림 2-2]에서 "E" 문자가 평문에서 가장 많이 사용되는 문자임을 알 수 있다. 그러므로 트루디는 "F" 문자가 "E" 문자로 대치되었다고 추측할 수 있다. 이러한 방법을 반복하면서 트루디는 단어를 인식할 때까지 시도를 계속할 수 있고 결국은 자신의 추측을 확신하는 단계에 도달하게 된다.

트루디가 처음에는 문장 속에서 공백의 위치를 모르기 때문에 결정하기 가장 쉬운 단어는 첫 단어일 것이다. 세 번째 문자가 "e" 이고 앞 두 문자의 빈도수가 높다면 트루디는 평문의 첫째 단어는 "the"라고 별 어려움 없이 추측할 수 있다. 나머지 암호문에도 이렇게 문자를 대치해 나가면 트루디는 더 많은 문자를 추측할 수 있고 따




**[그림 2-3] 암호문 영어 문자 빈도수 그래프**

라서 이 수수께끼는 신속하게 해결될 것이다. 트루디가 문제를 풀면서 오류를 범할 수도 있으나 통계에 대한 상식을 이용해 문제를 해결하면 450만년보다는 훨씬 빨리 평문을 찾을 수 있을 것이다.

단순치환 암호에 대한 이러한 공격에서 볼 수 있듯이 키 공간이 크다고 해서 충분한 보안을 제공하는 것은 아님을 알 수 있다. 이 공격에서 알 수 있는 또 다른 사실은 암호체계를 설계하는 사람은 이러한 교묘한 공격에도 대비해야 한다는 것이다. 그러면 이러한 교묘한 새로운 공격이 항상 새롭게 연구되고 있는 상황에서, 이런 유사한 모든 공격에 대비할 수 있는 방법이 있는가? 그 대답은 "없다"이다. 결과적으로, 암호체계는 그 암호체계에 대한 공격 방법이 발견되지 않는 동안에만 안전하다고 할 수 있다. 암호체계는 보다 실력이 뛰어난 암호 전문가의 공격을 견디어낼 수록 더욱 더 신뢰를 받을 수 있다.

### 2.3.3 안전성에 대한 정의

안전한 암호체계에 대한 타당한 정의는 다양하다. 이상적인 것은 그 체계에 대한 적합한 공격 방법이 없음을 수학적으로 증명하는 것이다. 그러나 그러한 증명은 지금까지 한가지 암호체계에서만 가능하였고 대부분의 암호체계에 대해서는 매우 비실용적인 것으로 알려져 있다.

암호체계의 강도에 대한 증명이 쉽지 않기 때문에 그에 차선으로 암호체계에 대한 최선의 공격 방법이 비현실적임을 검토하는 것이 타당하다. 이것이 가장 바람직한 특성으로 생각되므로, 약간 다른 정의를 채택하고자 한다. 즉, 어떤 암호체계에 대해 알려진 최선의 공격방법이 전수키조사 방법뿐일 때 그 암호체계는 **안전성**이 있다고 정의한다. 이 정의에 의하면 적은 수의 키를 가진 안전한 암호체계는 많은 수의 키를 가진 안전하지 않은 암호체계보다 깨어지기가 쉽다. 만약 암호화 알고리즘의 공격에

지름길이 있다면, 키 길이로 안전성의 수준을 "공시"하는 것은 이러한 정의에 비추어 볼 때 옳지 않음을 알 수 있다.

실질적으로, 안전하고 강한 암호체계를 선정하기 위해 이러한 정의를 바탕으로 안전한 암호체계를 선정하되, 전수키조사를 하기에는 비현실적일 정도로 큰 크기를 가지는 키 공간을 확보해야 한다. 이러한 안전성과 키 공간이라는 두 가지 요소 모두 필수적인 요소가 된다.

### 2.3.4 이중전위 암호

이중전위 암호로 암호화하려면 먼저 평문을 주어진 크기의 행렬 형태로 배열하고 행과 열을 기술된 순서에 따라 바꾼다. 예를 들어, 평문 attackatdawn을 3×4의 행렬로 정리하면 다음과 같다.

$$\begin{bmatrix} a & t & t & a \\ c & k & a & t \\ d & a & w & n \end{bmatrix}$$

만약 행을 $(1,2,3) \rightarrow (3,2,1)$로 바꾸고, 열을 $(1,2,3,4) \rightarrow (4,2,1,3)$으로 순서를 바꾸면 다음과 같은 형태가 된다.

$$\begin{bmatrix} a & t & t & a \\ c & k & a & t \\ d & a & w & n \end{bmatrix} \longrightarrow \begin{bmatrix} d & a & w & n \\ c & k & a & t \\ a & t & t & a \end{bmatrix} \longrightarrow \begin{bmatrix} n & a & d & w \\ t & k & c & a \\ a & t & a & t \end{bmatrix}$$

그러면 마지막 행렬이 암호문이 되는데, 여기서는 다음과 같은 결과가 나온다.

$$\texttt{NADWTKCAATAT} \tag{2-2}$$

이중전위 암호에서 키는 행렬의 크기와 열/행의 순열이다. 키를 알고 있는 암호문 수령자가 암호문을 적정한 크기의 행렬로 재정리하고 평문을 얻기 위해 순열을 역으로 수행하면 된다.

예를 들어, 암호문 (2-2)를 복호화하기 위해 먼저 암호문을 $3 \times 4$ 행렬로 정리한다. 이어서 열에 $(4,2,1,3)$으로 번호를 매기고 $(1,2,3,4)$로 재정렬한다. 그리고 다시 행에 $(3,2,1)$으로 번호를 매기고 $(1,2,3)$으로 재정렬하면 다음과 같은 결과가 나온다.

$$\begin{bmatrix} N & A & D & W \\ T & K & C & A \\ A & T & A & T \end{bmatrix} \longrightarrow \begin{bmatrix} D & A & W & N \\ C & K & A & T \\ A & T & T & A \end{bmatrix} \longrightarrow \begin{bmatrix} A & T & T & A \\ C & K & A & T \\ D & A & W & N \end{bmatrix}$$

결국 다시 복호화한 평문은 attackatdawn이 된다.

단순치환 암호와 달리 이중전위 암호는 메시지에 나타나는 문자를 감추지 않는다. 그러나 이 암호는 평문의 통계적 특성이 암호문 전체에 걸쳐 흩어져버렸기 때문에 평문 속에 포함된 통계적 정보에 의존하는 공격을 막을 수 있다.

이중전위 암호의 해독이 쉽지 않기 때문에 암호문에 평문 정보를 섞어버리는 이러한 아이디어는 현대 블록 암호체계에 유용하게 사용된다.

### 2.3.5 일회성 암호

버넘 암호(Vernam cipher)라고 불리는 일회성 암호는 안전성이 증명된 암호체계이다. 이 암호체계는 역사적으로 다양하게 사용되었지만 대부분 아주 실용적이지 못했다. 어쨌든 이 암호체계는 뒤에서 다시 설명할 몇 가지 중요한 개념을 잘 설명해 준다.

설명을 단순화하기 위해 알파벳 문자를 8개로 국한해 생각해 보자. 이러한 알파벳 문자와 해당 문자의 이진수 표현은 [표 2-1]과 같다. 이들 문자와 특정 비트의 매핑 자체는 비밀이 아니라는 것을 명심하기 바란다. 이러한 매핑은 ASCII 코드를 사용하는 목적과 유사한 역할을 한다. 앨리스라는 스파이가 다음과 같은 평문 메시지를 일회성 암호 방법으로 암호화한다고 가정하자.

| 문자 | e | h | i | k | l | r | s | t |
|---|---|---|---|---|---|---|---|---|
| 이진수 | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

**[표 2-1] 단축 알파벳**

앨리스라는 스파이가 다음 평문 메시지를 일회성 암호 방법으로 암호화한다고 가정하자.

```
heilhitler
```

앨리스는 먼저 [표 2-1]을 활용해 문자를 다음과 같은 비트열로 변환한다.

001 000 010 100 001 010 111 100 000 101

일회성 암호에는 암호화하고자 하는 메시지와 동일한 길이를 갖는 무작위 비트열로 만들어진 키가 필요하다. 암호문은 생성된 키와 평문을 XOR 연산한 결과로 생성된다. 평문과 키의 비트를 모듈로 2 방식으로 덧셈을 하는 것은 정말 멋진 일이다.

비트 $x$와 비트 $y$간의 XOR 연산은 $x \oplus y$로 표시한다. 그런데 $x \oplus y \oplus y = x$이기 때문에 복호화는 같은 키를 암호문과 XOR 연산하면 된다.

예를 들어, 스파이 앨리스가 다음과 같은 키를 가지고 있다고 가정하자.

111 101 110 101 111 100 000 101 110 000

이 키는 앞서 제시한 암호화될 메시지와 길이가 같다. 앨리스는 암호화를 위해 다음과 같은 계산을 할 것이다.

| | h | e | i | l | h | i | t | l | e | r |
|---|---|---|---|---|---|---|---|---|---|---|
| 평문: | 001 | 000 | 010 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| 키: | 111 | 101 | 110 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| 암호문: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| | s | r | l | h | s | s | t | h | s | r |

결과로 나온 암호문 비트를 문자로 다시 변환하면 최종적으로 전송할 암호문은 `srlhssthsr`이 된다.

스파이 밥은 앨리스로부터 다음과 같은 메시지를 수신해 다음과 같이 똑같은 키를 이용해 암호문을 복호화해 원래의 평문을 얻게 된다.

| | s | r | l | h | s | s | t | h | s | r |
|---|---|---|---|---|---|---|---|---|---|---|
| 암호문: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| 키: | 111 | 101 | 110 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| 평문: | 001 | 000 | 010 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| | h | e | i | l | h | i | t | l | e | r |

몇 가지 시나리오를 가정해 보자. 먼저 스파이 조직 내에 앨리스에게 찰리라는 적이 있다고 가정하자. 찰리가 앨리스의 메시지를 암호화하기 위해 사용한 실제 키는 다음과 같다고 주장한다고 하자.

101 111 000 101 111 100 000 101 110 000

밥이 찰리가 제시한 키로 암호문을 복호화하면 그 결과는 다음과 같다.

| | s | r | l | h | s | s | t | h | s | r |
|---|---|---|---|---|---|---|---|---|---|---|
| 암호문: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| “키”: | 101 | 111 | 000 | 101 | 111 | 100 | 000 | 101 | 110 | 000 |
| “평문”: | 011 | 010 | 100 | 100 | 001 | 010 | 111 | 100 | 000 | 101 |
| | k | i | l | l | h | i | t | l | e | r |

그러면 밥은 이 암호를 잘 이해하지 못하고 내용을 알아보기 위해 앨리스를 호출할 것이다.

이번에는 다른 시나리오를 생각해 보자. 앨리스가 자신의 암호문을 도청한 적에게 체포되었다고 가정하자. 도청자들은 보안성이 매우 강한 이 메시지를 해독하기 위해 앨리스에게 키를 제공하라고 강요할 것이다. 앨리스는 자신이 이중 간첩이라고 주장하며 이를 증명하기 위해 자신이 사용한 키가 다음과 같다고 주장할 수 있다.

111 101 000 011 101 110 001 011 101 101

앨리스를 체포한 사람이 앨리스가 제시한 키로 암호문을 복호화하면 다음과 같은 결과를 얻게 된다.

| | s | r | l | h | s | s | t | h | s | r |
|---|---|---|---|---|---|---|---|---|---|---|
| 암호문: | 110 | 101 | 100 | 001 | 110 | 110 | 111 | 001 | 110 | 101 |
| “키”: | 111 | 101 | 000 | 011 | 101 | 110 | 001 | 011 | 101 | 101 |
| “평문”: | 001 | 000 | 100 | 010 | 011 | 000 | 110 | 010 | 011 | 000 |
| | h | e | l | i | k | e | s | i | k | e |

결국 앨리스를 체포하였으나 암호를 잘 알지 못하는 그들은 오히려 앨리스의 애국심을 칭송하며 앨리스를 석방하게 될 것이다.

이러한 예는 일회성 암호가 증명된 안전한 암호체계인 이유를 잘 설명한다. 만약 키가 임의로 만들어 졌다면 암호문을 보는 공격자는 평문 메시지의 길이 이외의 정보는 파악할 수 없다. 즉, 주어진 암호문에서 같은 길이의 어떤 “평문”도 적정한 “키”를 선정하면 만들어질 수 있다. 가능한 평문의 확률은 모두 동일하다. 암호화하기 전에 임의의 문자를 메시지에 추가할 수도 있기 때문에 메시지의 길이도 의미가 없다. 그러므로 암호문은 평문에 대해 어떤 정보도 담고 있지 않는데, 이것이 바로 일회성 암호가 안전한 암호체계임을 증명한다.

물론, 암호문이 정확하다는 가정이 필요하다. 패드 또는 키는 임의로 생성되어야 하며, 한 번만 사용되어야 하고, 송신자와 수신자가 공유하고 있어야 한다.

안정성이 증명된 암호체계인 일회성 암호보다 더 나은 암호체계는 없다. 따라서 항상 일회성 암호체계만을 사용해야 한다고 생각할 수 있다. 그러나 일회성 암호에도 한 가지 심각한 단점이 있다. 즉, 키인 패드는 메시지 길이와 동일해야 하며 그 패드는 암호문이 복호화되기 전에 수신자에게 안전하게 전달되어야 한다는 것이다. 만약 그 패드를 수신자에게 안전하게 전달할 수 있다면 암호문을 생산할 필요없이 똑같은 전달 방법으로 평문 자체를 전달하는 편이 낫지 않을까? 뒤에서는 이러한 심각한 제한에도 불구하고 실질적으로 일회성 암호체계를 사용하는 것이 적절한 것으로 판단되는 사례를 볼 테지만, 데이터가 초고속으로 처리되는 현대 시스템에서 일회성 암호체계는 아주 비실용적이다.

일회성 암호가 한 번 밖에 사용될 수 없는 이유는 무엇일까? 두 개의 평문 $P_1$, $P_2$가 $C_1 = P_1 \oplus K$, $C_2 = P_2 \oplus K$로 각각 암호화되었다고 가정하자. 즉, 두 개의 메시지가 같은 "일회성" 패드인 $K$로 암호화되어 있다. 암호분석 분야에서 이런 것을 **깊이**(depth)가 있다고 표현한다. 깊이가 있는 일회성 암호에서는 두 개의 암호문을 다음과 같이 더하면 키가 사라져 버린다.

$$C_1 \oplus C_2 = P_1 \oplus K \oplus P_2 \oplus K = P_1 \oplus P_2$$

암호분석자인 트루디 외에는 이러한 경우가 생기는 것을 좋아하는 사람이 없을 것이다.

깊이가 있는 일회성 암호에 대한 구체적인 예를 들어보자. [표 2-1]의 비트와 문자 변경표를 사용해 다음과 같이 문자를 비트로 표현해 보자.

$$P_1 = \texttt{like} = 100\,010\,011\,000 \quad \text{and} \quad P_2 = \texttt{kite} = 011\,010\,111\,000$$

이때 모두 같은 키인 $K$ = 110 011 101 111으로 암호화한다고 가정한다. 그러면 첫 번째 암호문은 다음과 같이 만들어진다.

| | l | i | k | e |
|---|---|---|---|---|
| $P_1$: | 100 | 010 | 011 | 000 |
| $K$: | 110 | 011 | 101 | 111 |
| $C_1$: | 010 | 001 | 110 | 111 |
| | i | h | s | t |

두 번째 암호문을 같은 방식으로 만들면 다음과 같다.

| | k | i | t | e |
|---|---|---|---|---|
| $P_2$: | 011 | 010 | 111 | 000 |
| $K$: | 110 | 011 | 101 | 111 |
| $C_2$: | 101 | 001 | 010 | 111 |
| | r | h | i | t |

만약 암호분석자인 트루디가 두 개의 메시지에 깊이가 있음을 알게 된다면 트루디는 암호문의 2번째와 4번째 문자가 동일하다는 것을 보고, 바로 평문 $P_1$과 $P_2$의 2번째와 4번째의 문자도 동일할 것임을 간파할 것이다. 그리고 더욱 심각한 것은 트루디가 메시지 $P_1$을 추측해 나가면서 $P_2$를 이용해 자신의 추측 결과를 점검할 수 있다는 점이다. $C_1$과 $C_2$만을 알고 있는 트루디가 다음과 같이 $P_1$을 대략적으로 추정하였다고 가정해 보자.

$$\text{추정 } P_1 = \texttt{kill} = 011\,010\,100\,100$$

그러면 트루디는 해당되는 키가 다음과 같다고 추정할 수 있다.

| | k | i | l | l |
|---|---|---|---|---|
| 추정 $P_1$: | 011 | 010 | 100 | 100 |
| $C_1$: | 010 | 001 | 110 | 111 |
| 추정 $K$: | 001 | 011 | 010 | 011 |

트루디는 이렇게 추정된 키 $K$를 다시 이용해 $C_2$를 다음과 같이 "복호화"할 수 있다.

| | | | | |
|---|---|---|---|---|
| $C_2$: | 101 | 001 | 010 | 111 |
| 추정 $K$: | 001 | 011 | 010 | 011 |
| 추정 $P_2$: | 100 | 010 | 000 | 100 |
| | l | i | e | l |

위에서 사용한 키 $K$로 복호화한 $P_2$는 의미 있는 단어가 아니므로 트루디는 $P_1$ 의 추측이 틀렸다는 것을 알 수 있다. 그러다가 결국 $P_1$ = like로 다시 새롭게 추측해 정확한 키 $K$를 획득하고 $P_2$ = kite를 복호화할 수 있을 것이다. 그리하여 트루디는 키와 두 문장에 대한 복호화 결과가 정확하다고 확신할 수 있게 된다.

### 2.3.6 VENONA 프로젝트

VENONA 프로젝트[241]는 일회성 암호가 실세계에서 직접 사용된 한가지 사례라는 점에서 매우 흥미롭다. 구 소련 스파이들이 미국에 입국할 때 일회성 암호의 키를 휴대했다. 그 스파이들은 중요한 메시지를 암호화할 때 이들 키를 사용했고, 암호화된 메시지는 모스코바로 바로 보내졌다. 이들 메시지는 당시 가장 민감한 스파이 작전을 함축하고 있었다. 특히 최초 원자폭탄 개발에 대한 비밀은 스파이 활동의 가장 중요한 목표였다. 로젠베르크, 앨저 히스와 현재까지 알려진 많은 다른 스파이가(아직도 알려지지 않은 스파이가 많다) VENONA 프로젝트에 관계된 것으로 알려지고 있다.

소련의 스파이들은 잘 훈련되어 있어서 키를 절대로 반복 사용하지 않았음에도 미국의 암호분석자들은 도청된 많은 암호 메시지를 해독했다. 그러면 안정성이 증명된 일회성 암호는 어떻게 해독할 수 있었을까? 사실은 패드를 생성하는 방법에 약점이 있어서 패드 자체가 반복적으로 만들어 졌다. 결과적으로, 많은 메시지에 깊이가 있는데 이것이 암호분석자들이 메시지를 해독할 수 있게 한 원인이 되었다.

[표 2-2]는 해독된 VENONA 메시지의 한 부분이다. 이 메시지는 데이비드 그린글래스와 그의 부인인 루스에 관한 것이다. 여기서 LIBERAL은 율리우스 로젠베르크이며,

[C\percent Ruth] 그녀의 남편은 [v] 육군에 징집되었지만 최전방으로 배치되지 않았음.
그는 기계공학 기술자이며 현재 뉴멕시코주 산타페시에 위치하고 있는 ENORMOUS
[ENORMOZ][vi] 공장에서 일하고 있음.
[그룹 45개는 해독 불가]
ENORMOUS 공장에서 일하고 있는 VOLOK [vii]을 구속하라. 그는 FELLOWCOUNTRYMAN
[ZEMLYaK] [viii] 조직원임. 그는 어제 그가 일자리에서 해고되었음을 인지하였음.
해고 사유는 진보적인 조직에서 활발하게 활동했던 그의 과거 경력임.
FELLOWCOUNTRYMAN 소속에서 LIBERAL은 CHESTER [ix]와 접촉 중임. 그들은 공작금
지급을 위해 매달 접선하고 있음. CHESTER는 우리가 공동 노력에 대해 만족하고
있는지, 또 잘못 이해하고 있는 부분이 없는지 관심이 많음.
그는 [KONKRETNAYa RABOTA] 업무의 세부사항에 대해 질문하지 않았음.
CHESTER가 LIBERAL 그룹의 역할에 대해 알고 있는 범위 안에서, ENORMOUS 및 다른
기술 분야에서 일하고 있는 사람 중 리더들에 대해 LIBERAL을 통해 C에게
문의하는 데 동의하도록 요구하였음.

**[표 2-2] 1944년 9월 21일 VENONA 해독 메시지**

그는 그의 부인 에씰과 함께 핵무기 관련 스파이 혐의로 처형되었다.[4] 소련 핵 폭탄의 코드명은 ENORMOUS로 추정된다. VENONA 암호에 관한 매우 흥미있는 읽을거리는 참고문헌 [241]번을 찾아보기 바란다.

### 2.3.7 코드북 암호

고전적인 코드북 암호는 단어와 해당 단어에 할당된 코드를 수록하고 있는 사전과 같다. [표 2-3]은 세계1차 대전 중 독일이 사용한 유명한 코드북의 일부를 발췌한 내용이다.

예를 들어, Februar라는 독어를 암호화하는 방법은 전체 단어를 5자리 숫자인 코드인 13605로 대치하면 된다. 암호화에는 [표 2-3]의 코드북이 사용되었고 복호화에는 각 단어와 함께 이들 단어에 해당되는 5자리 숫자가 오름차순으로 정리된 코드북이 사용된다. 코드북은 치환 암호의 일종이지만 여기서의 치환은 전체 단어나 절을 바꾸는 것이므로 단순치환과는 차이가 있다.

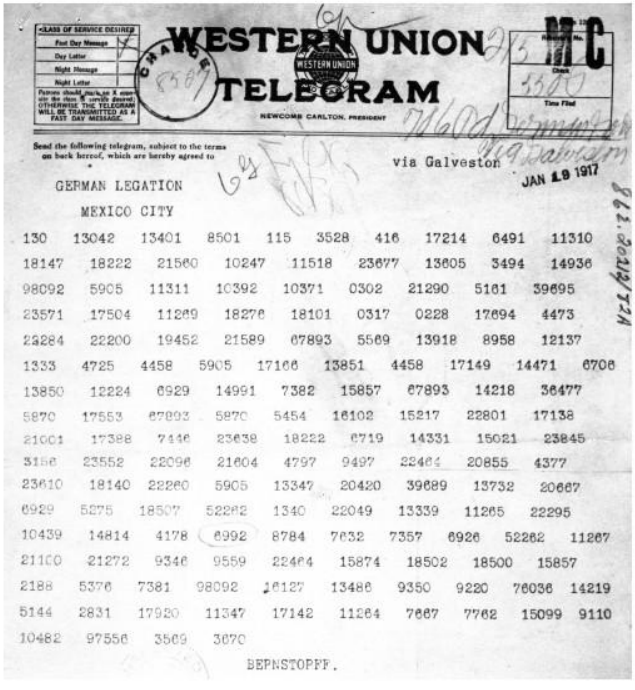
[표 2-3]에서 설명한 코드북은 그 유명한 짐머맨 전보를 암호화할 때 사용되었다. 1917년 독일 외무부 장관인 아더 짐머맨이 암호화된 전문을 멕시코시에 있는 독일 대사관에 전송했다. 그 암호화된 전문은 [그림 2-4][174]와 같으며, 이 전문은 영국

[4] 데이비드 그린글래스는 원자폭탄 스파이 죄목으로 15년 형을 선고받았으며 10년간만 수감되었다. 그는 에씰 로젠버르크가 이 사건과 관련된 정도에 대한 결정적인 증언에 거짓말을 하였다고 나중에 주장했다. 아마도 그 증언이 그녀가 사형 선고를 받게 된 결정적 요인이 되었을 것이다.

| 평문 | 암호문 |
|---|---|
| Februar | 13605 |
| fest | 13732 |
| finanzielle | 13850 |
| folgender | 13918 |
| Frieden | 17142 |
| Friedenschluss | 17149 |
| ⋮ | ⋮ |

**[표 2-3] 독일 코드북에서 발췌**

에게 도청당했다. 그 당시 영국과 프랑스는 독일과 전쟁중인 우방국이었고, 미국은 중립을 지키고 있었다[238].



WESTERN UNION TELEGRAM

NEWCOMB CARLTON, PRESIDENT

Send the following telegram, subject to the terms on back hereof, which are hereby agreed to

via Galveston

JAN 19 1917

GERMAN LEGATION
MEXICO CITY

130 13042 13401 8501 115 3528 416 17214 6491 11310
18147 18222 21560 10247 11518 23677 13605 3494 14936
98092 5905 11311 10392 10371 0302 21290 5161 39695
23571 17504 11269 18276 18101 0317 0228 17694 4473
24284 22200 19452 21589 07893 5569 13918 8958 12137
1333 4725 4458 5905 17166 13851 4458 17149 14471 6706
13850 12224 6929 14991 7382 15857 67893 14218 36477
5870 17553 67893 5870 5454 16102 15217 22801 17138
21001 17388 7346 23638 18222 6719 14331 15021 23845
3156 23552 22096 21604 4797 9497 22464 20855 4377
23610 18140 22260 5905 13347 20420 39689 13732 20667
6929 5275 18507 52262 1340 22049 13339 11265 22295
10439 14814 4178 6992 8784 7632 7357 6926 52262 11267
21100 21272 9346 9559 22464 15874 18502 18500 15857
2188 5376 7381 98092 16127 13486 9350 9220 76036 14219
5144 2831 17920 11347 17142 11264 7667 7762 15099 9110
10482 97556 3569 3070

BEPNSTOPFF.

**[그림 2-4] 짐머맨 전보 원문**

러시아는 독일의 손상된 코드북을 복원해 이것의 일부를 영국에 넘겨주었다. 영국은 힘들게 분석한 결과 짐머맨 전보[62]를 복호화하기 위한 코드북을 완전히 복원하게

되었다. 그 전보에는 독일 정부가 "무제한적인 잠수함전"을 시작하려는 계획과 이러한 계획은 결국 미국과의 전쟁을 피할 수 없는 결과를 초래할 것이라는 내용을 담고 있었다. 결과적으로, 짐머맨은 미국에 대항해 싸우기 위해 멕시코를 독일의 연합군으로 끌어들여야 한다고 결심했다. 독일이 멕시코에게 연합하는 대가로 제시한 것은 멕시코의 옛 영토인 "텍사스, 뉴멕시코, 아리조나"를 멕시코에 돌려주는 것이었다. 복호화된 짐머맨 전보가 미국에 노출된 후 미국의 민심이 독일에 적대적으로 돌아섰으며 정기 여객선 루시타니아호가 격침되자 결국 미국도 독일에 전쟁을 선포하게 되었다.

영국은 독일이 그들의 암호가 노출되었음을 알아채고 그 암호체계 사용을 중단할 것을 우려해 짐머맨 전보를 공개하는 것을 처음에는 망설였다. 그러나 영국의 암호분석자는 짐머맨 전보와 비슷한 시간대에 암호화되지 않은 채 케이블로 전송되고 있는 짐머맨 전보와 유사한 전보들을 발견했다. 영국은 시중에 돌아다니는 암호화되지 않은 내용과 똑같이 각색하여 짐머맨 전보를 공개하였다. 이러한 상황을 분석한 독일은 그들의 코드북이 노출된 것이 아니라는 결론을 내리고, 1차 세계대전 내내 그 코드북을 사용하여 중요한 메시지를 암호화했다.

현대 블록암호는 평문에서 암호문(또는 암호문에서 평문)을 생산할 때 복잡한 알고리즘을 사용한다. 그러나 더 높은 차원에서 보면 블록암호는 각 키마다 다른 코드북을 사용하는 하나의 코드북으로 볼 수 있다.

### 2.3.8 1876년 선거에서의 암호

1876년의 미국 대통령 선거는 그야말로 막상막하의 접전이었다. 그 당시에 미국 국민들의 마음 속에 남북전쟁에 대한 기억이 생생히 남아있는 상태에서, 남부의 주들에 대한 합중국 재편입이 급진적으로 진행되고 있었기 때문에 국론이 여러 측면에서 심각하게 분열되어 있었다.

그 선거에서 경선자는 공화당 러더퍼드 헤이즈와 민주당 새뮤얼 틸덴이었다. 틸덴이 전체 국민들의 투표에서 상대보다 득표가 약간 앞서고 있었으나 대통령은 선거인단이 결정하도록 되어 있었다. 선거인단은 각 주에서 대의원단을 파견하고, 그 전체 대의원단이 그 주에서 가장 많이 득표한 후보에게 투표하도록 되어 있다(각 대의원이 특정 후보에게 반드시 투표하도록 법으로 강제하지는 않았지만 대의원이 다른 후보에게 투표하는 경우는 거의 없다).

1876년에 4개 주의 선거인단 대의원들이 논쟁 중이었고 상호 주장이 대등했다. 15명으로 구성된 위원회가 어떤 대의원이 합법적인가를 결정하도록 권한이 부여되었고, 결국 이 위원회가 대통령을 결정하는 셈이 되어 버렸다. 그 위원회는 4개 주 모두

| 평문 | 암호문 |
|---|---|
| Greenbacks | Copenhagen |
| Hayes | Greece |
| votes | Rochester |
| Tilden | Russia |
| telegram | Warsaw |
| ⋮ | ⋮ |

**[표 2-4] 1876년 선거 코드북**

헤이즈에게 투표하는 것이 적법하다고 결정해 결국 그가 미국의 대통령으로 선출되었다. 그러자 틸덴 지지자들은 헤이즈 진영에서 대의원들에게 뇌물을 주어 부정하게 투표하도록 했다고 고소했으나 표면적으로 나타난 증거는 아무 것도 없었다.

선거가 끝나고 수 개월 후, 틸덴지지자들이 당시 논쟁 중이었던 주들의 대의원들에게 보낸 수많은 암호문을 기자들이 발견했다. 사용된 암호체계는 단어들을 전위시키고 부분적인 코드북을 함께 사용하는 방식이었다. "핵심" 단어에만 코드북을 적용하고 전위방법은 정해진 길이에 해당하는 메시지에 고정된 순열을 적용했다. 사용할 수 있는 메시지의 길이는 10개, 15개, 20개, 25개, 30개 단어이었고, 모든 메시지는 이러한 길이에 맞도록 만들어 졌다. [표 2-4]는 이 코드북의 일부이다.

10개 단어로 된 메시지의 순열은 다음과 같다고 해보자.

$$9,3,6,1,10,5,2,7,4,8$$

다음 메시지는 10개 단어로 된 실제 암호화 메시지의 하나이다.

```
Warsaw they read all unchanged last are idiots can't situation
```

전위와 치환을 원위치로 하여 위 메시지를 해독한 "바르샤바" 전문은 다음과 같다.

```
Can't read last telegram. Situation unchanged. They are all idiots.
```

이런 식의 취약한 암호체계에 대한 암호분석은 상대적으로 용이하다[96]. 정해진 길이에 대한 순열을 반복적으로 사용하기 때문에 특정 길이의 많은 메시지는 코드북 뿐만 아니라 순열에 있어서도 깊이를 가진다. 그러므로 암호분석자는 부분적인 코드북에 대한 정보가 없어도 같은 길이의 모든 메시지를 비교할 수 있고, 상대적으로 쉽게 고정 순열을 알아낼 수 있다. 분석가는 주어진 같은 길이의 모든 메시지가 같은 순열을 사용하고 있을 가능성을 예측할 수 있는 지혜를 가져야 하고, 이러한 통찰력

있으면 그 순열은 쉽게 발견될 수 있다. 코드북은 문맥 간의 의미나 암호 메시지의 내용에 대한 실마리를 제공하는 암호화되지 않은 메시지를 이용해 추정할 수 있다.

이렇게 복호화된 메시지가 밝혀낸 것은 무엇일까? 틸덴의 지지자들이 논쟁 중이었던 주 대의원들에게 뇌물을 제공하려고 하였다는 사실의 폭로는 기자들의 흥미를 자아내기에 충분했다. 아이러니한 것은 틸덴 진영이 헤이즈 진영이 하였다고 고소한 바로 그 죄목이 바로 틸덴 진영 자신들의 죄목이라는 점이다!

누가 봐도 이 암호체계는 아주 엉성하게 설계되었고 보안성이 약하다. 여기서 얻을 수 있는 한 가지 교훈은 키 재사용(또는 남용)은 암호분석자에게는 아주 유용한 암호체계의 결함이 된다는 사실이다. 앞에서 살펴본 예의 경우, 순열이 매번 재사용되었는데 이러한 결과는 암호분석자에게 대조할 수 있는 정보를 더 많이 제공해 그 순열이 밝혀지게 했다. 현대 암호체계에서는 단일 키 사용을 제한해 임호분석자가 개별 키에 대한 많은 정보를 축적하지 못하도록 하고 있으며, 이를 통해 키가 노출되는 위험을 줄인다.

## 2.4 현대 암호 역사

어제가 오늘을 너무 많이 잡아먹지 않도록 하라.

*Abraham Lincoln*

20세기 전체에 걸쳐 암호학은 세계적인 많은 사건에서 중요한 역할을 해왔다. 또한 20세기 후반에는 암호학이 상업과 사업 분야에서 정보교환 수단의 중요한 기술이 되었다. 짐머맨 전보는 지난 세기 후반에 암호분석이 가졌던 정치적, 군사적 역할을 보여준 첫 번째 예이다. 이 절에서는 추가적으로 지난 세기에 있었던 암호분석과 관련된 흥미있는 역사적 사건을 언급하고자 한다. 흥미진진한 암호학의 역사에 대한 자료는 칸의 책[122]를 참조하기 바란다.

1929년에, 미국의 국방장관인 헨리 스팀슨은 미국 정부의 공식적인 암호분석 활동을 "신사는 타인의 우편물을 읽지 않는다[225]"라는 불멸의 문구와 함께 종식시켰다. 이러한 공식적인 암호분석 활동의 중단은 일본의 진주만 공격에 이르러 비로소 값비싼 댓가를 지불해야 하는 실수였음이 증명되었다.

1941년 12월 7일 진주만 공격 직후, 미국은 암호분석 프로그램을 본격적으로 재가동시켰다. 제 2차 세계대전 기간 중에 연합군의 암호분석자의 성공은 놀랄만 했으며, 이 기간은 암호분석의 "황금시대"로 불린다. 사실상 암호체계의 주축은 모두 무너졌고 이러한 암호체계로부터 입수된 정보의 가치는 엄청난 것으로 판명되었다.

태평양 지역 전장에서는 퍼플(Purple) 암호라고 불리는 암호체계가 일본 정부 고위층의 통신수단으로 사용되었다. 이 암호는 미국의 암호분석자들이 일본의 진주만 공격

이전에 이미 해독했다. 그러나 이 암호해독으로부터 획득된 정보(코드명 MAGIC)는 곧 일어날 공격에 대한 확실한 징후를 제공하지는 못했다[63]. 일본제국의 해군은 미군이 이미 해독한 JN-25로 알려진 암호를 사용하고 있었다. JN-25로부터 획득된 정보는 산호해와 미드웨이에 걸친 광범위한 전투에서 열세인 미국이 초기에 일본군의 태평양 진출을 저지할 수 있도록 결정적으로 기여했다. 일본 해군은 이 전투에서 입은 손실을 끝까지 회복하지 못했다.

한편, 유럽에서는 이니그마 암호(Enigma Cipher, 코드명 ULTRA) 해독이 역시 제 2차 세계대전 중에 연합군에게 결정적인 도움을 주었다[78, 90]. ULTRA 정보는 너무 중요했기 때문에 1940년 11월에 윈스턴 처칠은 영국 코벤트리시에 대한 독일 공군의 임박한 공격을 인지하였음에도 불구하고 그 정보의 주 출처가 이니그마 암호 해독에서 획득되었다는 이유로 이를 공개하지 않았다[231]. 처칠은 독일군의 공격에 대한 경고 행위가 독일이 자신의 이니그마 암호가 해독되었다는 사실을 감지하게 될 수 있다는 것을 염려했기 때문이다[5]. 독일이 그들의 암호가 해독되었음을 눈치채지 못하도록 하면서 ULTRA에서 생산된 정보를 잘 활용하는 것은 전쟁 중에 끊임없이 부딪히는 난제였다[37].

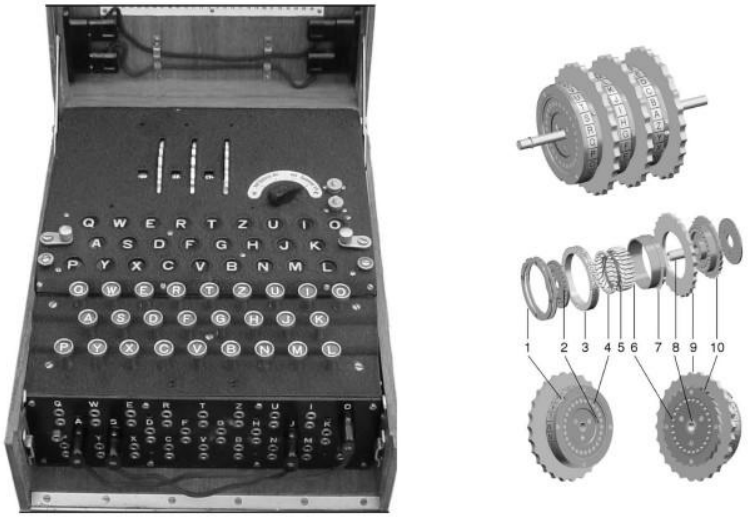
이니그마 암호는 폴란드인에 의해 최초로 해독되었다. 폴란드가 함락된 후에 폴란드 암호분석자들은 프랑스로 탈출했다. 곧이어 프랑스도 나치에 의해 함락되자 그들은 다시 영국으로 탈출해 그곳에서 영국 암호학자들에게 그들의 지식을 제공하게 되었다. 특이한 점은 영국에서 폴란드 암호학자들이 이니그마 암호를 연구하는 것이 더 이상 허락되지 않았다는 것이다[6]. 그러나 컴퓨터 분야 선구자인 알랜 튜링을 포함한 영국의 암호분석팀은 팀 내부적으로 보다 향상된 이니그마 공격 방법을 지속적으로 개발하였다[78]. [그림 2-5]는 이니그마 암호 장비와 회전자의 구조를 보여주는데, 내부 동작 방법은 6장의 연습문제 12번을 참조하기 바란다.

제 2차 세계대전 후 암호학은 감춰진 예술에서 현실상의 과학으로 전환되었다. 이 때 1949년에 발표된 클로드 섀넌의 논문 **보안체계의 정보 이론**[209]이 전환점이 되었다고 볼 수 있다. 섀넌은 이 논문에서 일회성 암호체계가 안전함을 증명하였고 암호체계 설계의 두 가지 기본 원칙인 **혼돈**(confusion)과 **확산**(diffusion) 이론을 제시했다.

암호체계를 설계함에 있어 혼돈은 평문과 암호문과의 상관 관계를 숨기는 반면 확산은 평문의 통계적 성격을 암호문 전반에 확산시켜 이를 숨기는 것이다. 단순치환 암호와 일회성 암호는 혼돈 특성만을 갖지만 이중전위 암호는 확산 특성만을 가진다.

[5]**역자주**: 저자는, "처칠이 코벤트리시에 대한 공격을 미리 경고하지 않았다"는 설은 잘못된 것이라고 정정했다.

[6]**역자주**: 저자는, 최초로 이니그마를 해독한 폴란드 암호분석자들은 프랑스가 함락된 이후에도 메세지를 해독하는 일을 계속하였다고 정정했다.




**[그림 2-5] 이니그마 암호 장비 및 사용된 회전자**

확산 특성만 사용한 일회성 암호가 안전한 암호체계임이 증명되었으므로 확산 특성은 자체만으로 안정성을 보장하기에 충분하다. 하지만 혼돈은 자체의 특성만으로는 안전성을 보장할 수 없다.

혼돈과 확산이라는 두 개념은 오늘날 암호체계 설계에도 적용되고 있다. 다음 장에서 이러한 중요한 개념이 현대 블록암호 설계에 어떻게 적용되는지 설명할 것이다.

최근까지 암호학은 정부가 관장하는 업무였다. 이것이 대용량의 전자 데이터를 보호해야 하는 컴퓨터 혁명으로 인해 1970년대에 이르러 극적인 변화를 가져오게 되었다. 1970년대 중반에 미 정부는 합법적이면서 상업적으로 안전한 암호체계가 필요함을 깨닫게 되었으나 그 당시 상용 암호체계는 미비했다. 미 국가표준국(NBS)[7]은 암호 알고리즘의 표준화 요구를 받게 되었다. 이러한 요구에 부응해 미 정부에서 공식적으로 사용할 DES라고 알려진 데이터 표준 암호 알고리즘이 탄생하게 되었다.

현대 암호역사에 있어 DES의 역할은 아무리 강조해도 지나치지 않다. 다음 장에서는 DES를 보다 자세히 살펴볼 것이다.

DES 암호 알고리즘이 발표된 후, 암호학에 대한 학문적 관심이 급속도로 커졌다. 공개키 암호가 DES 발표 직후에 등장했다. 보다 정확히 말하면 재발견된 것이라고 할

[7]미 국가표준국(NBS, National Bureau of Standards)은 NIST(National Institute of Standards and Technology)의 새로운 이름이다. 미 정부 기관의 명칭을 3글자로 줄여 명명하기 위해 명칭을 바꾼 것 같다.

수 있다. 1980년대에는 높은 수준의 암호 관련 논문들이 발표된 암호 학술대회가 매년 개최되었다. 1990년대에는 암호칩(Clipper Chip)과 오래된 DES를 대신할 수 있는 두 종류의 알고리즘이 개발되는 성과가 있었다.

정부는 암호학과 이와 관련된 분야의 정부 조직에 대해 예산지원을 계속했지만, 암호 관련 업무 및 연구는 이제 정부의 손에서 벗어나 민간부분으로 깊숙이 파고 들고 있었다.

## 2.5 암호학의 분류

다음 3개 장에서는 암호의 중요한 세 가지 주제인 **대칭키 암호체계, 공개키 암호체계, 해시함수**를 자세히 살펴본다. 그러므로 여기서는 먼저 이런 주제들에 대해 간략히 언급하고자 한다.

공개키와 대칭키의 차이가 미미해 보일 수 있으나 공개키 암호체계는 대칭키 암호로는 얻을 수 없는 몇 가지 유용한 기능을 제공한다.

공개키 암호체계에서는 암호화 키를 공개할 수 있다. 예를 들어, 어떤 사람이 인터넷에 자신의 공개키를 공개한다면 인터넷에 접속한 사람 누구나 그 암호화 키에 대해 사전에 협조하지 않고도 그 사람에게 메시지를 암호화해 전송할 수 있다. 이것은 암호화를 위해 키에 대해 사전에 협조해야 하는 대칭키와의 명백한 차이점이다. 공개키를 사용하기 이전에는 대칭키를 안전하게 전달하는 것이 현대 암호에 있어 유일한 약점이었다. 대칭키 암호 분배체계의 문제점을 적나라하게 보여준 한 가지 실례는 워커 가족 간첩망이 저지른 사건이다. 이는 워커 가족이 미군들이 사용하는 암호키를 소련에 팔아넘겼다는 사실이 거의 20년이 지나서야 발각된 사건이다.[61, 72].

공개키 암호체계에는 대칭키 암호와는 비교할 수 없는 매우 유용한 특징이 또 하나 있다. 하나의 메시지가 공개키 대신 개인키로 "암호화"되었다고 가정하자. 공개키는 공개되어 있어 누구나 이 메시지를 복호화할 수 있다. 따라서 이러한 방식의 암호화는 별 의미 없어 보인다. 그러나 이 기능은 손으로 쓴 서명과 유사하게 디지털 형식의 서명으로 사용될 수 있다. 이 서명은 누구나 읽을 수는 있으나 오직 서명한 사람만이 서명을 만들 수 있다는 점에서 매우 유용하다. 이러한 주제와 관계된 내용은 추후 **전자서명**을 다룰 때 보다 자세히 설명할 것이다.

대칭키 암호로 할 수 있는 기능을 공개키 암호체계에서도 모두 할 수 있을 뿐 아니라 대칭키 체계에서는 불가능한 기능도 수행할 수 있다. 그렇다면 대칭키 없이 모두 공개키 암호체계를 사용하면 되지 않을까? 대칭키가 많이 사용되는 주된 이유는 바로 처리 속도 때문이다. 따라서 오늘날 방대한 데이터에 대한 암호화에는 주로 대칭키

암호체계가 사용된다. 또한 공개키 암호체계도 현대 정보보안에서 결정적인 역할을 수행하고 있다.

앞에서 설명한 고전 암호는 모두 대칭키 암호체계이다. 현대의 대칭키 암호는 **스트림암호**(stream cipher)와 **블록암호**(block cipher)로 세분화될 수 있다. 스트림암호는 일회성 암호를 일반화한 것으로, 안전성에 대한 희생을 어느 정도 감수하면서 적당한 수준에서 키의 길이를 한정시킨다. 한편, 블록암호는 코드북을 일반화한 것이다. 블록암호에서는 키가 코드북을 결정한다. 즉 키가 고정되는 한 같은 코드북이 사용되며 반대로 키가 변하면 다른 코드북이 선정된다.

제 2차 세계대전 기간 중에는 스트림암호가 주로 사용되었으나 오늘날에는 특별한 경우를 제외하고는 블록암호가 대칭키 암호의 대표적인 암호로 사용된다. 일반적으로 블록암호는 소프트웨어로 구현하는 것이 용이한 반면 스트림암호는 하드웨어를 통해 보다 효율적으로 구현된다.

세 번째로 해시함수를 살펴보자. 이 함수는 크기에 관계없이 데이터를 입력받아서 어떤 고유 특성을 만족시키는 고정된 크기의 출력 데이터를 만들어 낸다. 여기서 요구되는 고유 특성이란 입력 데이터가 한 비트 이상 변경되면 출력 데이터는 절반 이상이 변경되어야 하다는 것이다. 또 다른 예는 임의의 두 입력 데이터가 동일한 출력 데이터를 만들어내는 경우가 거의 없어야 하는 것이다. 이러한 특성을 만족시키는 함수가 충분히 유용한지 또는 그러한 함수가 실제로 존재하는지는 명확하지 않을 수 있으나 실제로 그러한 함수가 존재하며 놀라울 만큼 다양한 문제에서 매우 효율적으로 활용되는 것으로 판명되었다는 사실을 확인할 것이다.

## 2.6 암호분석의 분류

암호분석의 목표는 평문이나 키를 찾아내는 것이다. 커크호프 원칙에 의하면, 암호분석자 트루디는 암호 알고리즘의 내부 작동에 대해 완전히 알고 있다고 가정한다. 또 다른 기본 가정은 트루디가 암호문을 획득할 수 있다는 것이다. 그렇지 않다면 왜 공들여 암호화하려고 하겠는가? 만약 트루디가 알고리즘과 암호문만을 알고 암호분석을 시도한다면 트루디는 **암호문**(Ciphertext) 공격을 수행해야 한다. 이것은 트루디 관점에서는 가장 불리한 공격 시나리오이다.

만약 트루디가 **알려진 평문**(known plaintext)에 대한 지식이 있다면 성공 가능성은 훨씬 높아질 것이다. 이는 트루디가 평문의 일부분에 해당하는 암호문을 알 수 있는 경우이다. 이러한 평문과 암호문의 쌍은 키에 대한 정보를 제공할 수도 있다. 모든 평문을 다 알 수 있다면 키의 복구는 의미가 없다. 그러나 트루디가 평문의 일부분만 획득(또는 추측)할 수 있는 경우는 많다. 예를 들어, 많은 종류의 데이터가 일정하게

형식화된 머리글을 포함하고 있다(이메일이 그 좋은 예이다). 만약 그러한 데이터가 암호화된다면 공격자는 머리글에 해당하는 평문의 일부분을 추측할 수 있고 그에 해당하는 암호문을 함께 볼 수 있게 된다.

트루디가 평문을 임의로 선택해 이에 해당하는 암호문을 볼 수 있는 경우도 있다. 이러한 공격 방법을 **선택된 평문**(chosen plaintext) 공격이라고 정의할 수 있다. 그러면 트루디는 평문을 어떻게 마음대로 선택할 수 있을까? 임의의 메시지를 전송하면 해당하는 암호문이 되돌아 오는 프로토콜도 있다. 그리고 트루디가 암호체계에 일시적으로 접근할 수 있어서 자신이 임의로 선택한 평문을 암호화해 볼 수 있는 경우도 상정할 수 있다. 예를 들어, 앨리스가 컴퓨터를 로그오프하는 것을 잊고 점심식사를 하러 나가는 경우가 있다면, 트루디는 앨리스가 돌아 오기 전에 자신이 선택한 임의의 메시지를 암호화해 전송할 수 있을 것이다. 이런 "점심시간(Lunchtime) 공격" 방식은 여러 가지 형태로 존재한다.

**적응적 선택된 평문**(adaptively chosen plaintext) 공격은 공격자에게 더 유리하다. 이 시나리오에서는 트루디가 평문을 선택해 암호문의 결과를 본 후에 그 암호문에 의거해 다음 평문을 선택한다. 경우에 따라서는 이러한 공격방법이 트루디의 일을 매우 간단하게 만들어 줄 수 있다. **관련 키** 공격 방법도 분야에 따라 매우 잘 활용될 수 있다. 이 아이디어는 키가 특별한 방법으로 생성되어 있는 시스템의 약점을 이용하는 것이다.

암호학자들이 주로 새로운 학술논문을 발표할 필요성을 느낄 때 다른 방식의 공격에 대해 고민하기도 한다. 암호체계는 어떤 경우에도 그에 대한 성공적인 공격이 존재하지 않을 경우에만 비로소 안전하다고 말할 수 있다. 다음 장에서 "성공적"이라는 말의 의미를 보다 구체적으로 정의할 것이다.

마지막으로, 공개키 암호체계에 적용되는 특별한 공격 시나리오를 소개한다. 트루디가 앨리스의 공개키로 암호화된 암호문을 도청하였다고 가정하자. 만약 트루디가 그 암호문의 평문이 "예" 또는 "아니오" 중의 하나라고 추측한다면, 그는 앨리스의 공개키로 추정되는 그 평문을 모두 암호화할 수 있다. 그리고 둘 중 하나가 도청한 암호문과 일치한다면 그 메시지는 해독된 것인데, 이러한 공격을 **순방향 탐색**이라고 한다. 이 후의 장에서는 순방향 탐색이 대칭키 암호에는 적용되지 않지만 해시함수를 공격하는 데 사용되는 경우를 살펴 볼 것이다.

공격자가 가능한 키를 모두 시도하는 것을 방지하기 위해 키 공간이 충분히 커야 한다고 설명한 바 있다. 순방향 탐색 공격은 공개키 암호에서 공격자가 모든 가능한 평문 메시지를 암호화할 수 없도록 하려면 평문 메시지 공간이 충분히 커야 함을 보여준다.

## 2.7 요약

이 장에서는 단순치환, 이중전위, 코드북 , 일회성 암호 등과 같은 다양한 고전 암호체계를 살펴보았으며, 앞으로 다룰 암호체계에 대한 중요한 요점을 설명했다. 그런 다음, 암호분석에 관한 몇 가지 기본적인 요소에 대해 설명했다.

다음 장의 주제는 현대적인 대칭키 암호체계이다. 공개키 암호, 해시함수 암호분석에 대한 주제를 다루는 장이 이어지는데, 특히 암호학은 이 책의 후반부에 보안 프로토콜을 다루는 장에서 다시 한번 중요한 주제로 다룰 것이다.

## 2.8 연습문제

1. 주어진 카이사르 암호가 사용된 다음 암호문을 복호화해 평문을 구하라.

   VSRQJHEREVTXDUHSDQWU

2. 다음 암호문에서 평문과 키를 구하라.

   CSYEVIXIVQMREXIH

   위에 주어진 암호문은 알파벳이 $n$만큼 이동한 단순치환 암호이다.

3. 초당 $2^{40}$개의 키를 검사할 수 있는 컴퓨터가 있다면 키 공간 $2^{128}$을 전수키조사 방식으로 키를 발견하는 데 몇 년이 소요될까?

4. 복합치환 암호(Vigenere 암호)가 작동하는 방식을 예를 들어 설명하라. 단순치환 암호의 통계적 공격을 사용해 Vigenere 암호를 공격하는 방법을 찾아보라.

5. 1876 선거 기간에 사용된 암호는 주어진 문장 길이에 항상 고정된 순열을 사용해 매우 취약했다. 이러한 취약점을 확인하기 위해 단어를 섞어 놓은 다음 문장에서 (1,2,3,…,10) 순열을 찾아라. 단, "San Francisco"는 한 단어이며 세 문장에 사용된 순열은 동일하다.

   first try try if you and don't again at succeed
   only you you you as believe old are are as
   winter was in the I summer ever San Francisco coldest spent

6. 부분적인 코드북과 단어들의 순열을 사용한 1876 선거에서 사용한 암호체계는 매우 취약했다. 이 암호체계를 더욱 안전하게 보완해 새로운 버전의 암호체계를 설계하라.

7. 암호학 측면에서 혼돈과 확산의 개념을 정의하라. 고전암호에서 혼돈만 적용된 암호와 확산만이 적용된 암호를 구분해 설명하라. 또 이 장에서 다룬 암호 중에서 혼돈과 확산이 모두 적용된 암호는 어떤 것일까?
8. (2-1)에 있는 단순치환 예제를 복호화하라.
9. 이 장 맨 처음에 등장하는 이상한 나라의 앨리스 인용구에 있는 암호문을 복호화하라(힌트: 그 메시지는 단순치환 방법으로 암호화되어 있다).
10. 단순치환 암호로 암호화된 다음 메시지를 복호화하라.

```
GBSXUCGSZQGKGSQPKQKGLSKASPCGBGBKGUKGCEUKUZKGGBSQEICA
CGKGCEUERWKLKUPKQQGCIICUAEUVSHQKGCEUPCGBCGQOEVSHUNSU
GKUZCGQSNLSHEHIEEDCUOGEPKHZGBSNKCUGSUKUASERLSKASCUGB
SLKACRCACUZSSzeUSBEXHKRGSHWKLKUSQSKCHQTXKZHEUQBKZAEN
NSUASZFENFCUOCUEKBXGBSWKLKUSQSKNFKQQKzeHGEGBSXUCGSZQ
GKGSQKUZBCQAEIISKOXSZSICVSHSZGEGBSQSAHSGKHMERQGKGSKR
EHNKIHSLIMGEKHSASUGKNSHCAKUNSQQKOSPBCISGBCQHSLIMQGKG
SZGBKGCGQSSNSZXQSISQQGEAEUGCUXSGBSSJCQGCUOZCLIENKGCA
USOEGCKGCEUQCGAEUGKCUSZUEGBHSKGEHBCUGERPKHEHKHNSZKGGKAD
```

11. 단순치환 암호에 대한 분석을 지원할 수 있는 프로그램을 작성하라. 프로그램은 암호문을 읽어 각 문자의 빈도수를 계산하고 분석자를 위해 그 결과를 전시해야 한다. 분석자가 키를 추측할 수 있어야 하며 추정되는 키를 사용해 복호화한 결과를 전시할 수 있어야 한다.
12. 연습문제 11번의 프로그램을 확장해 메시지를 우선 복호화할 수 있도록 하라. 프로그램 작성에 활용할 수 있는 한 가지 현명한 방법은 다음과 같다. 처음 키를 추측할 때는 알려진 알파벳의 빈도수와 계산된 빈도수를 활용하라. 그리고 추정해 복호화한 결과에서 사전에 있는 단어의 수를 계산하고 이를 점수화해 사용하라. 다음 단계로, 키에 있는 각 문자를 빈도수가 높은 인접한 문자와 바꾸어 점수를 다시 계산하라. 만약 점수가 올라가면 키를 그 문자로 바꾸고 그렇지 않으면 그대로 두라. 이런 작업을 점수가 더 이상 올라가지 않을 때까지 모든 알파벳 문자를 사용해 반복하라. 수작업을 하는 암호분석자를 지원하기 위해 연습문제 11번에 있는 프로그램의 모든 기능은 그대로 유지해야 한다.
13. 4행×4열로 된 이중전위 암호를 사용해 다음 메시지를 암호화하라.

```
we are all together
```

열 순열은 다음과 같다.

$$(1,2,3,4) \longrightarrow (2,4,1,3)$$

행 순열은 다음과 같다.

$$(1,2,3,4) \longrightarrow (3,1,2,4)$$

14. 다음 암호문을 복호화하라.

```
IAUTMOCSMNIMREBOTNELSTRHEREOAEVMWIH
TSEEATMAEOHWHSYCEELTTEOHMUOUFEHTRFT
```

위 메시지는 7행×10열로 구성된 행렬을 이용한 이중전위 방법으로 암호화되어 있다(힌트: 평문의 첫 단어는 "there"이다).

15. 행렬의 크기를 안다고 가정하고 이중전위 암호에 대한 자동화된 공격 방법을 개략적으로 기술하라.

16. [표 2-1]에 나오는 코드화된 문자들을 이용해 다음 두 암호문은 동일한 일회성 암호를 사용해 암호화되었다.

```
KHHLTK ,   KTHLLE
```

각 메시지에 대해 가능한 평문과 대응하는 일회성 암호를 구하라.

17. [표 2-1]의 코드화된 문자를 이용해 다음 암호문은 일회성 암호로 암호화되어 있다.

```
KITLKE
```

평문이 "thrill"이라면 키는 무엇일까? 평문이 "tiller"라면 키는 무엇일까?

18. 한 메시지가 1,024비트로 구성되어 있다고 가정하자. 64비트의 키를 1,024비트로 확장하는 방법을 설계하라. 확장된 1,024 비트를 일회성 암호에서 하는 것과 동일하게 메시지와 XOR 연산을 하라. 만들어진 암호문은 일회성 암호만큼 안전성이 있을까? 일회성 암호만큼 안정성이 보장된 암호가 가능할까?

19. 코드북 암호의 전산화된 버전을 설계하라. 설계된 암호체계는 가능한 많은 코드북을 포함해야 하며, 각각의 코드북은 특정한 메시지를 암호화(또는 복호화)할 때 사용될 코드북을 결정하기 위한 키를 가지고 있어야 한다.

20. 이 장에서 순방향 탐색 공격이 공개키 암호체계를 어떻게 공격하는가를 설명했다. 동일한 방법이 대칭키 암호를 공격하는 방법으로 사용될 수 없는 이유는 무엇일까?