5. Feistel 암호 구성요소

Feistel 암호는 **3가지** 타입의 구성요소를 가진다.

〈Feistel 암호가 가지는 3가지 타입의 구성요소〉

① 자기 자신을 역(inverse)으로 갖는 것
② 역함수가 존재하는 것
③ 역함수가 존재하지 않는 것

- Feistel 암호는 역이 존재하지 않는 구성요소를 결합하고,
- 암복호 알고리즘에서 **동일한 구성요소**를 사용한다.

① 자기 자신을 역으로 갖는 것

암복호 과정에서 **동일한 키**를 사용하면, **xor** 연산은 **자기 자신을 역**으로 갖는다.

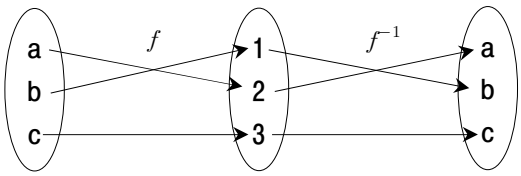
// 예제 : 역연산 – xor 연산의 **역**

```
    0101 ← 평문 p
xor)1001 ← 키(암호키), 암호화 과정
    1100 ← 암호문 c
xor)1001 ← 키(복호키), 복호화 과정
    0101 ← 평문 p
```

- 암복호 알고리즘에서 **동일한 키**를 사용하므로 두 알고리즘은 서로 **역(inverse)** 관계이다.
- c1=c2이면, p1=p2이다.

② 역함수가 존재하는 것

Feistel 구조에서 암복호화는 서로 **역관계**이므로 **역함수가 존재하는 것**을 사용하는 것은 당연하다.



- **예** : 입출력 수가 같은 단순 P-box는 역함수가 존재한다.(**3비트**를 **입력**받아, 3**비트**를 **출력**)

③ 역함수가 존재하지 않는 것

- Feistel 암호는 역이 존재하지 않는 구성요소를 사용한다.
- **역이 존재하지 않는 구성요소**로 설계된 암복호 알고리즘이 어떻게 **서로 역 관계**가 될 수 있는가?
- **배타적 논리합**(xor, ⊕) 연산을 이용하여, **역이 존재하지 않는 구성요소**를 **역 관계**를 만들 수 있다.

// 예제 : 함수 f(key)의 기능이 다음과 같을 때, 평문은 1010이고, key가 101일 때, 암복호 과정은?

함수 f(key)의 기능	• key의 첫 번째와 세 번째의 비트를 추출한다. • 추출한 두 비트를 10진수로 간주하고, 제곱한다. • 제곱한 값을 4bit 2진수로 변환한다.

- 함수 f(key)는 역함수가 존재하지 않는다.(**3비트**를 **입력**받아, **4비트**를 **출력**하므로)

↓

↓**풀이**

↓

함수 f(key)	암복호 과정	암복호 과정을 수식으로 표현	
↓			
함수 f(101)		**암호**	C = P ⊕ f(key) = 1010 ⊕ 1001 = 0011
↓			
11	1010 ← 평문 p		
↓	xor)1001 ← 키(암호키), 암호화 과정		
3	0011 ← 암호문 c		
↓	xor)1001 ← 키(복호키), 복호화 과정	**복호**	P = C ⊕ f(key) = 0011 ⊕ 1001 = 1010
3^2	1010 ← 평문 p		
↓			
9			
↓			
1001			

- C = P ⊕ f(key)에서 "P ⊕ f(key)"를 **혼합기**(mixer)라 한다.
- 함수 f(key)는 역함수가 존재하지 않지만
- 혼합기 "P ⊕ f(key)"는 자기 자신을 역함수로 가진다. - **xor** 연산의 **특징**
- 해서, Feistel 암호는 역이 존재하지 않는 구성요소를 사용할 수 있다.

↓

↓**결론**

↓

역이 존재하지 않는 구성요소로 설계된 암복호 알고리즘이 서로 역 관계가 될 수 있다.