Feistel Structure

SeoulTech Cryptography & Information Security Lab.

석병진

2019-07-19

Contents

1. Feistel Structure

Feistel Structure?

- 블록암호 구조의 설계 방식 중 하나.
- 1970년대 초 Horst Feistel에 의해 제안됨.

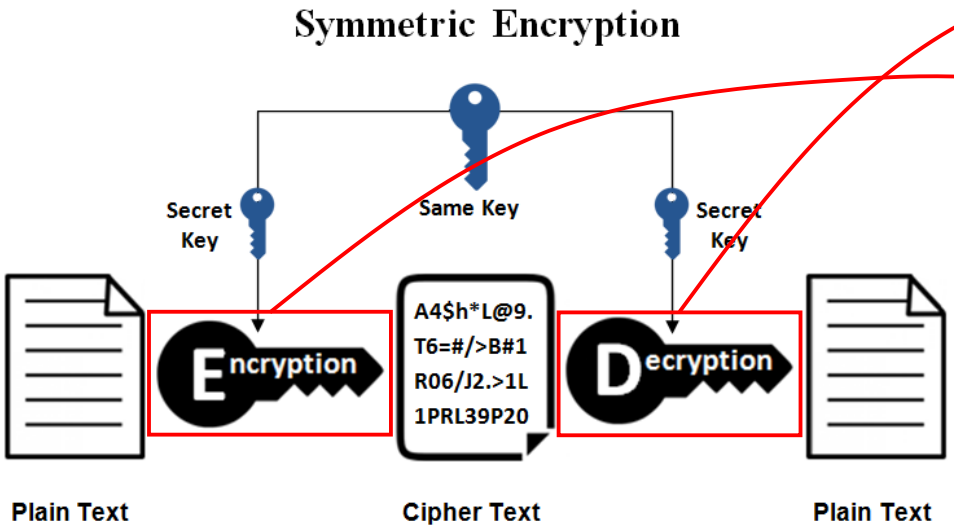


Figure Ref: https://www.ssl2buy.com/wiki/symmetric-vs-asymmetric-encryption-what-are-differences

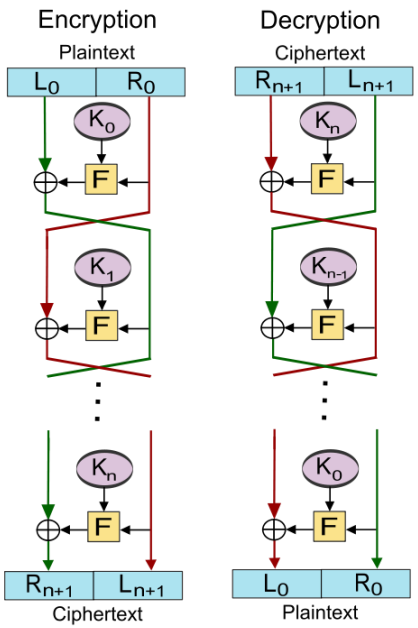


Figure Ref: https://en.wikipedia.org/wiki/Feistel_cipher

1. Feistel Structure

Enc/Dec process

- **암호화 과정**
  - 평문 m을 (L_0, R_0)로 분할.
  - $i = 0,1, \ldots, r$에 대해 다음을 수행.
    - $L_{i+1} = R_i$
    - $R_{i+1} = L_i \oplus F(R_i, k_{i+1})$
  - $c = (L_{r+1}, R_{r+1})$을 암호문으로 출력.

- **복호화 과정**
  - 암호문을 입력으로 평문을 출력.
  - 기본적으로 암호화 과정과 동일.
  - 단, 라운드 키를 역순으로 사용함.

- **마지막 라운드는 단순히 자리를 바꿈.**

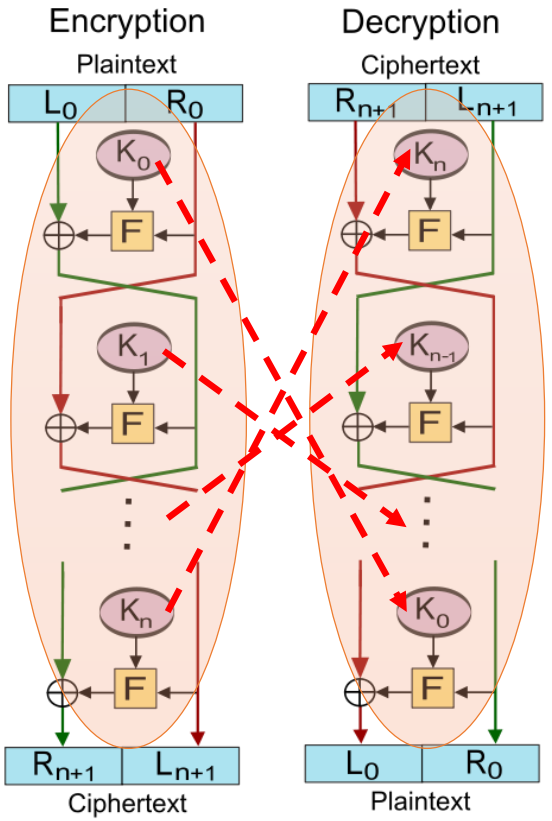


Figure Ref: https://en.wikipedia.org/wiki/Feistel_cipher

1. Feistel Structure

Features

- **F 함수가 가역일 필요가 없음.**
  - ✓ F 함수는 복호화 과정에 영향을 주지 않음.

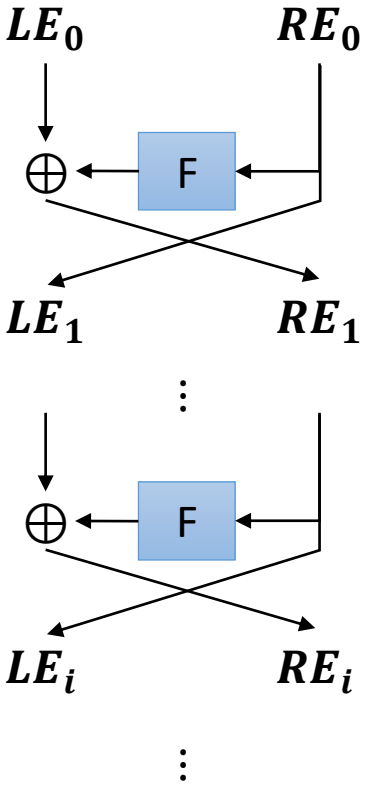
- **이론적 안전성 증명이 상대적으로 용이함.**
  - ✓ Why? 한 라운드가 단 두줄의 수식으로 표현됨(매우 간단함) → 수학적 분석이 용이함.

- **암호화/복호화 과정이 동일함(구현시 함수 재사용 가능).**
  - ✓ 최근 IoT 와 같은 경량 환경에서 좋은 전략으로 평가받아 많은 최신 경량 암호에서 채택하고 있음.
  - ✓ SIMON, LEA 등에서 사용.

1. Feistel Structure

Proof - Decryption

- 16라운드를 통해 암호화되는 암호로 가정
- 암호화 과정
  - 입력 평문: $(\boldsymbol{LE_0}, \boldsymbol{RE_0})$, 암호화 중간값: $(\boldsymbol{LE_i}, \boldsymbol{RE_i})$
- 복호화 과정
  - 입력 암호문: $(\boldsymbol{LD_0}, \boldsymbol{RD_0})$, 암호화 중간값: $(\boldsymbol{LD_i}, \boldsymbol{RD_i})$



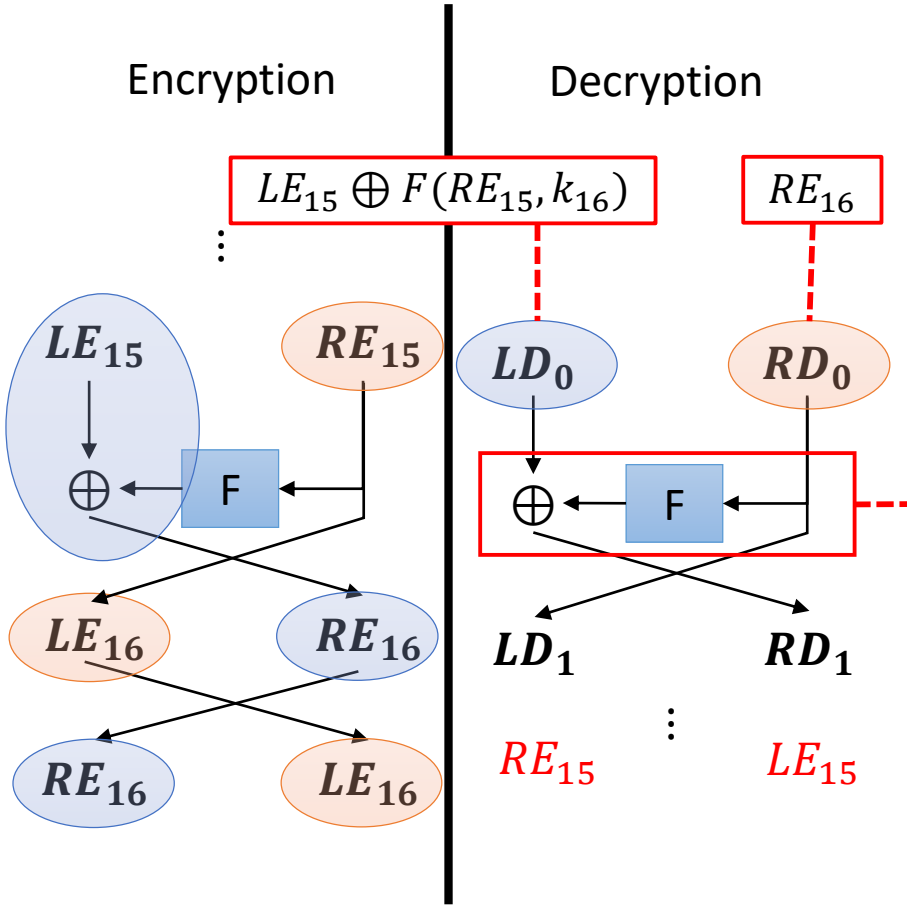
1. Feistel Structure

Proof – Decryption (1/2)

- **16라운드를 통해 암호화되는 암호로 가정**
- **암호화**
  - 입력 평문: $(\boldsymbol{LE_0}, \boldsymbol{RE_0})$, 암호화 중간값: $(\boldsymbol{LE_i}, \boldsymbol{RE_i})$
- **복호화**
  - 입력 암호문: $(\boldsymbol{LD_0}, \boldsymbol{RD_0})$, 암호화 중간값: $(\boldsymbol{LD_i}, \boldsymbol{RD_i})$
- **Proof**
  - 최종 암호문: (RE_{16}, LE_{16}).
  - $LE_{16} = RE_{15}$
  - $RE_{16} = LE_{15} \oplus F(RE_{15}, k_{16})$
  - $LD_1 = RD_0 = LE_{16} = RE_{15}$
  - $RD_1 = LD_0 \oplus F(RD_0, k_{16}) = RE_{16} \oplus F(RE_{15}, k_{16}) = [LE_{15} \oplus F(RE_{15}, k_{16})] \oplus F(RE_{15}, k_{16}) = LE_{15}$

1. Feistel Structure

Proof – Decryption (2/2)



- $LE_{16} = RE_{15}$
- $RE_{16} = LE_{15} \oplus F(RE_{15}, k_{16})$
- $LD_1 = RD_0 = LE_{16} = RE_{15}$
- $RD_1 = LD_0 \oplus F(RD_0, k_{16}) = RE_{16} \oplus F(RE_{15}, k_{16}) = [LE_{15} \oplus F(RE_{15}, k_{16})] \oplus F(RE_{15}, k_{16}) = LE_{15}$

$[LE_{15} \oplus F(RE_{15}, k_{16})] \oplus F(RE_{15}, k_{16})$

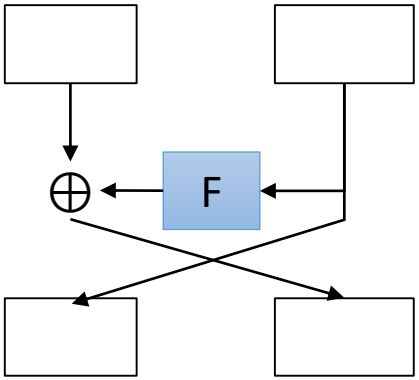
계속 반복하고 마지막 라운드 자리바꿈을 하면 원래의 평문이 구해짐.

1. Feistel Structure

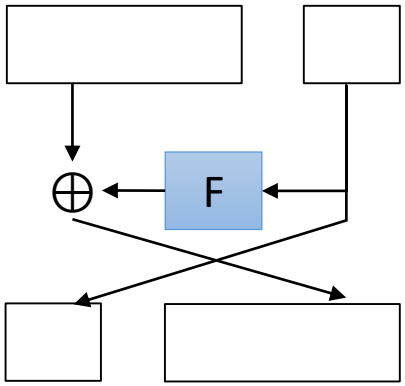
Quiz. SPN 구조와 대비해 Feistel 구조의 장단점은?

2. Modified Feistel Structures

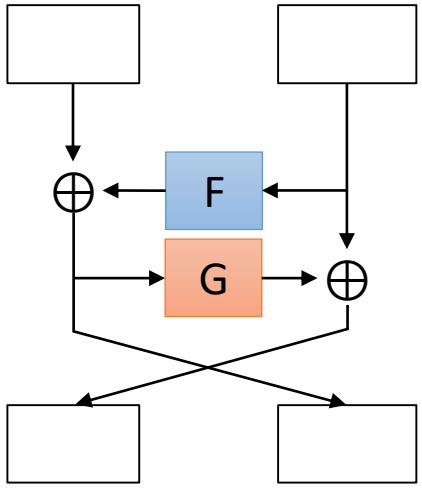
Balanced Feistel (Classical)



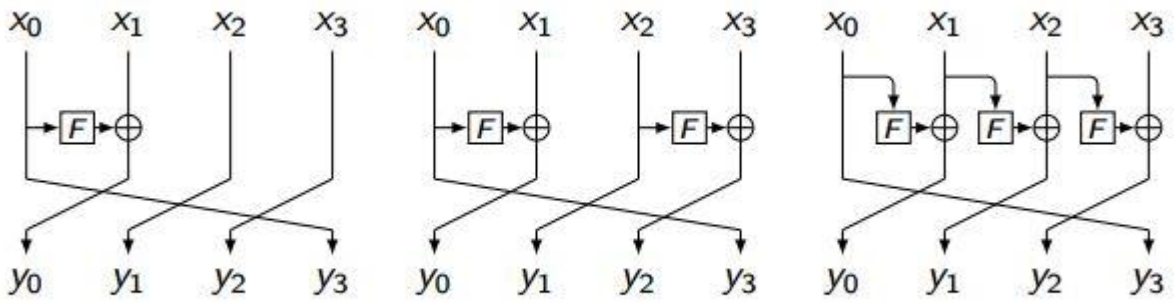
Unbalanced Feistel



Alternating Feistel



Generalized Feistel



Type-1 (CAST-256, Lesamnta) Type-2 (HIGHT, CLEFIA) Type-3

Figure Ref: http://cryptowiki.net/index.php?title=Generalized_Feistel_networks

3. ARX Design Paradigm

ARX (Addition-Rotation-Xor)

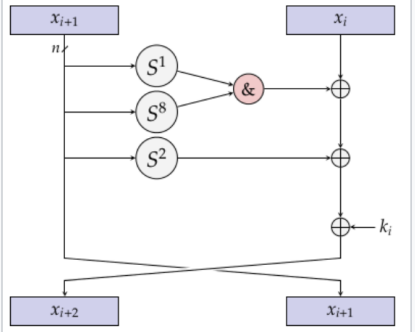
- **블록암호의 연산을 modular addition, rotation, xor 만으로 구성하는 방식**
- **최근 자원이 제한된(Resource-constrained) 경량 환경을 위해 ARX-based 암호가 인기를 끌고 있음.**
- **블록암호 TEA, XTEA, Threefish, HIGHT, LEA, Speck, CHAM 등에서 사용됨.**
- **더불어 스트림 암호, 해시함수에서도 ARX 기반으로 설계된 암호 함수들이 존재함.**
  - 스트림 암호: Salsa20, ChaCha, HC-128 등
  - 해시함수: BLAKE, Skein, Cubehash 등

- **장점**
  - PC 환경에서 고속으로 동작
  - 알고리즘 명시와 이해, 구현이 쉬움
  - Timing attack 에 안전함.

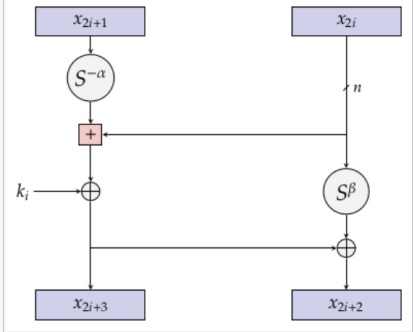
- **단점**
  - 차분, 선형 공경에 대한 안전성 증명이 어려움.
  - 부채널 공격에 대응시 효율성 저하가 심함.

3. ARX Design Paradigm

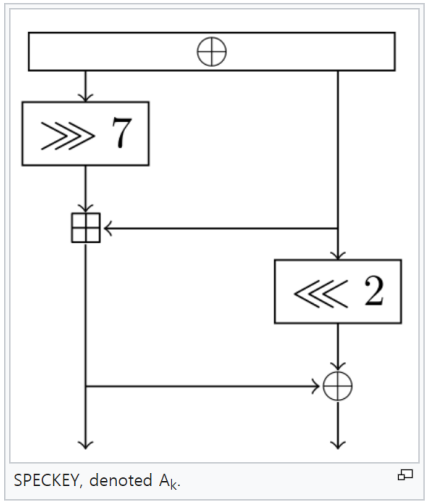
Examples



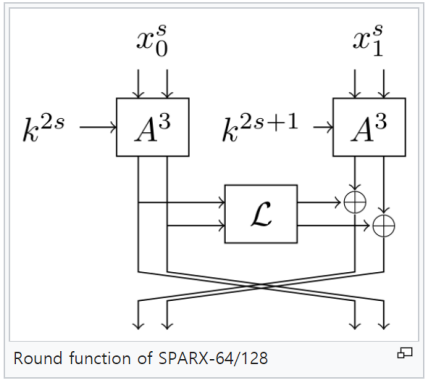
Round function of SIMON



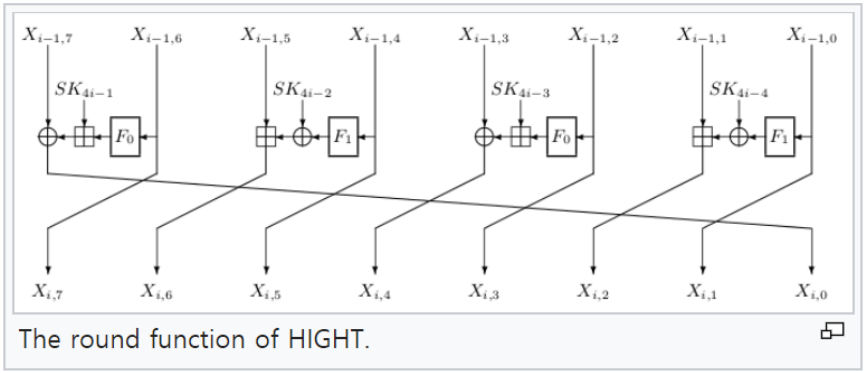
Round function of SPECK



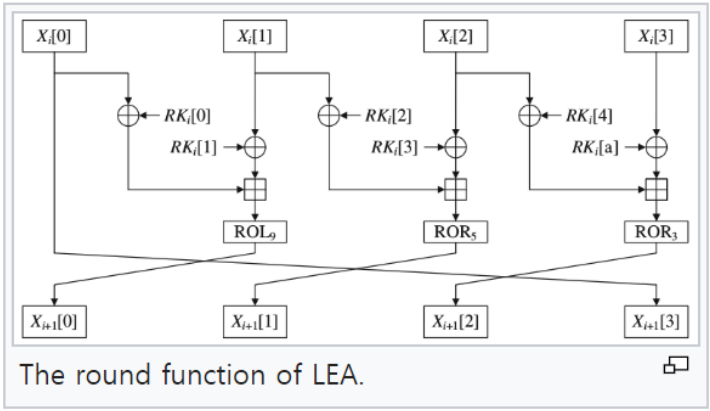
SPECKEY, denoted A_k.



Round function of SPARX-64/128



The round function of HIGHT.



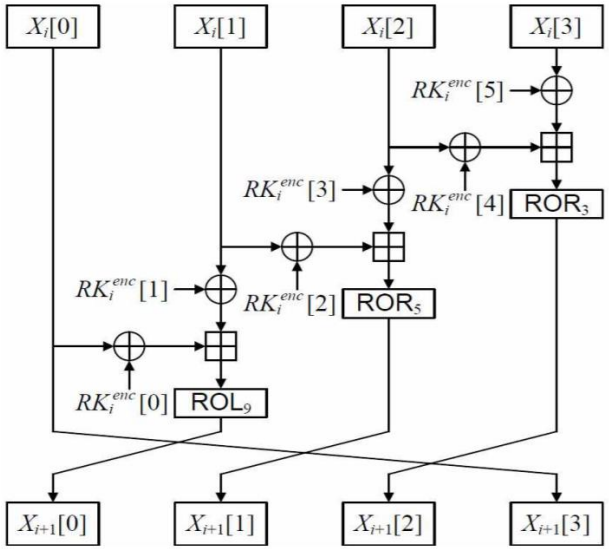
The round function of LEA.

Figure Ref: https://www.cryptolux.org/index.php/Lightweight_Block_Ciphers

4. Home work: LEA

LEA 구현

- 블록암호 LEA는 2013년에 개발된 국내 표준 블록 암호.
- ARX 기반 GFN 구조.
- 다음 링크에서 규격서 참고 가능
  - ✓ https://seed.kisa.or.kr/kisa/skill/EgovLeaInfo.do



(그림 5-2) 암호화 과정의 i번째 라운드 함수 ($0 \le i \le (Nr-1)$)

감사합니다~