DEFAULT 구현

장경배

https://youtu.be/tqR2gb-pWW0

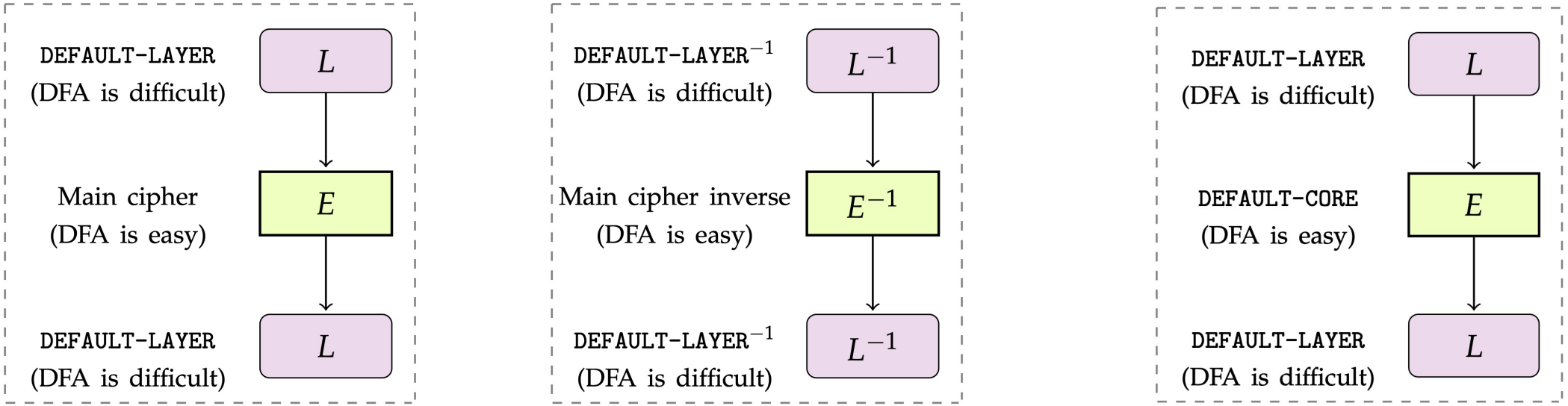
DEFAULT

- **Differential Fault Attack**
  - Fault Attack에서 가장 많이 사용되는 공격
  - 매우 강력한 공격이며, 기존 공격에 안전했던 대부분의 암호들 모두 취약성을 보임
  - 최근 몇 년 동안 이에 대한 DFA에 대한 대응책은 나왔지만
    - DFA에 대한 내성을 가지도록 설계된 암호는 없음

- **DEFAULT : Cipher Level Resistance against Differential Fault Attack**
  - DFA(Differential Fault Attack). 내성을 갖도록 설계된 대칭키 암호

DEFAULT

- 제안하는 구조는 대칭키 암호의 앞, 뒤에 **Protection layer를 연결하여 DFA에 대한 검색 복잡도를 증가시킴**
  - **DFA Protection layer의 핵심은 선형 구조의 Sbox**
  - 일반적으로, 선형 구조의 Sbox는 차분 공격에 취약하기 때문에 암호 설계에서 사용되지 않음
- 본 논문에서는, **선형 구조의 SBox가 DFA에 대한 보호기능을 제공**하면서
  사실, 우수한 **암호화 성능도 제공하는 절충안**이 될 수 있다고 주장 → DEFAULT로 검증



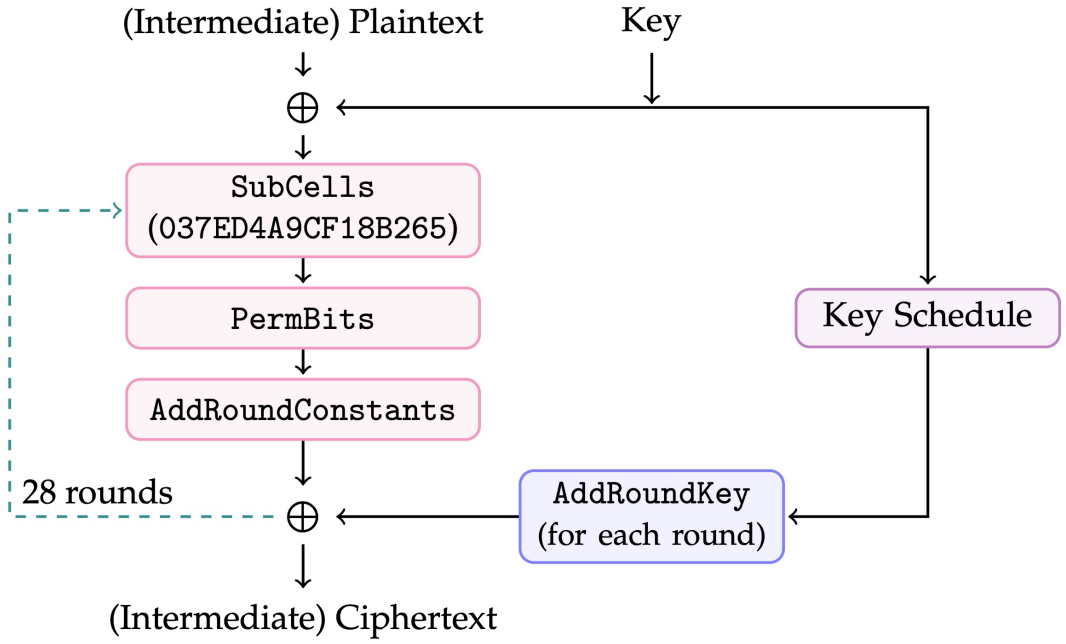
(a) Encryption **(b)** Decryption

Figure 8.2: Sandwiched construction for DEFAULT

DEFAULT Layer

- **28 Round의 SPN 구조**
- **Sbox = 037ED4A9CF18B265**

$$
\begin{aligned}
y_0 &= x_0 \oplus x_1 \oplus x_2, \\
y_1 &= x_0 \oplus x_1 \oplus x_0x_1 \oplus x_0x_2 \oplus x_1x_3 \oplus x_2x_3, \\
y_2 &= x_1 \oplus x_2 \oplus x_3, \\
y_3 &= x_0x_1 \oplus x_2 \oplus x_0x_2 \oplus x_3 \oplus x_1x_3 \oplus x_2x_3.
\end{aligned}
$$



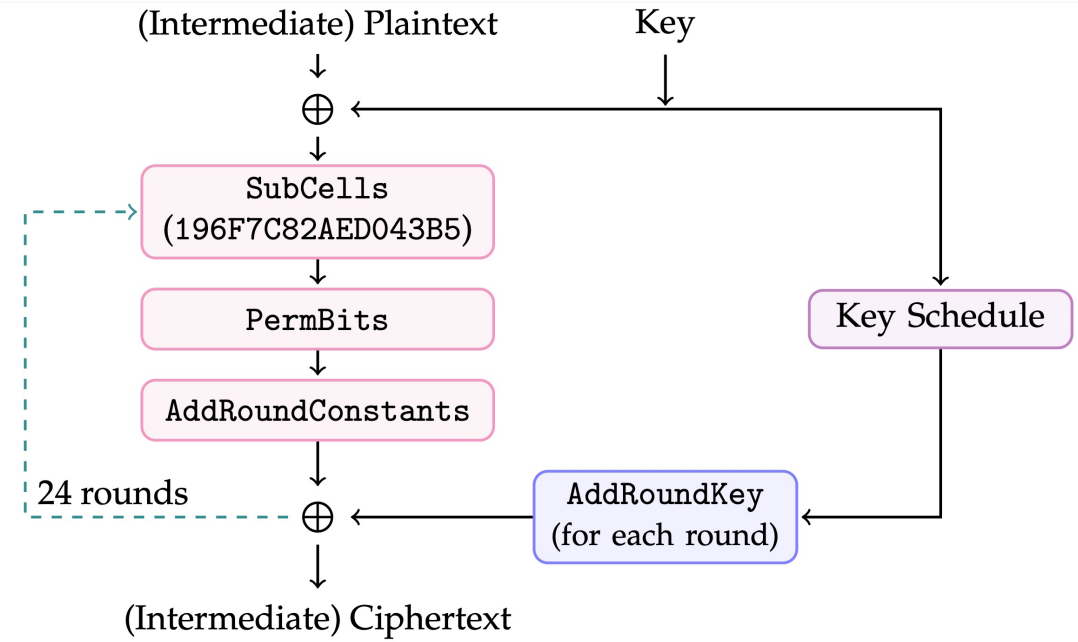
(a) DEFAULT-LAYER

- GIFT-128에서 사용되는 **Permutation Table을 동일하게 사용**
- GIFT-128과 유사한 **AddRoundConstants**
- 해당 버전(학위 논문)에서 키 스케줄은 **DEFAULT Core의 키 스케줄을 따름** (ASIACRYPT 버전과 다름)
- **AddRoundKey는 128-bit 라운드 키 전부 XOR**

DEFAULT Core

- **24 Round의 SPN 구조**
- **Sbox = 196F7C82AED043B5**

$$
\begin{aligned}
y_0 &= 1 \oplus x_1 \oplus x_0x_1 \oplus x_0x_2 \oplus x_3, \\
y_1 &= x_1 \oplus x_2 \oplus x_0x_2 \oplus x_3, \\
y_2 &= x_1 \oplus x_2 \oplus x_0x_3, \\
y_3 &= x_0 \oplus x_1x_2 \oplus x_3 \oplus x_0x_3 \oplus x_0x_1x_3 \oplus x_2x_3.
\end{aligned}
$$



(b) DEFAULT-CORE

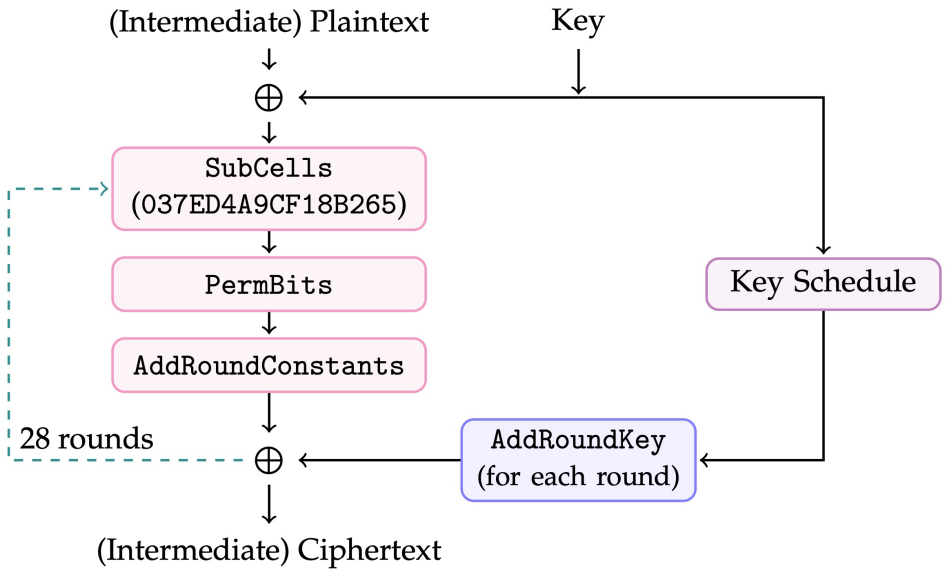
- GIFT-128에서 사용되는 **Permutation Table을 동일하게 사용**
- GIFT-128과 유사한 **AddRoundConstants**
- **Rotation 연산으로 구성 된 키 스케줄** →

$$
\begin{aligned}
k_7 \parallel k_6 \parallel \cdots \parallel k_0 &\leftarrow (k_7 \parallel k_6 \parallel \cdots \parallel k_0) \ggg 20 \\
k_7 &\leftarrow k_7 \ggg 1
\end{aligned}
$$

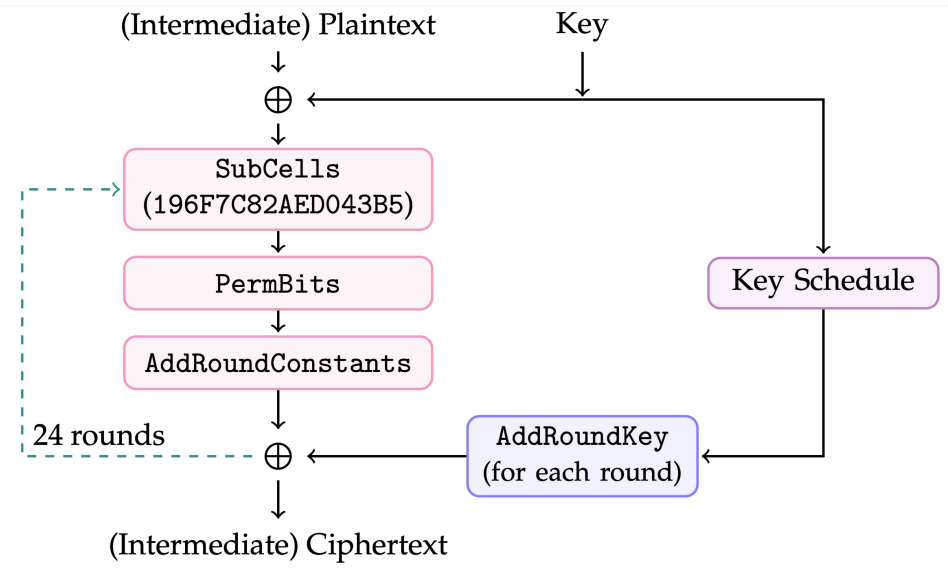
Implementation

Key Schedule and `AddRoundKey`

`DEFAULT-LAYER` does not have a key schedule of its own. Instead, the round keys can be extracted from key schedule of the main cipher for ease of implementation. While for the choice of the key schedule of `DEFAULT-LAYER`, we use that of `DEFAULT-CORE`. In the `DEFAULT` construction, to avoid two consecutive round key additions (for efficiency), we skip the that at the final round of the initial `DEFAULT-LAYER` and the initial round of the final `DEFAULT-LAYER`.



(a) `DEFAULT-LAYER`



(b) `DEFAULT-CORE`

Implementation

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
$P_{128}(i)$	0	33	66	99	96	1	34	67	64	97	2	35	32	65	98	3	4	37	70	103	100	5	38	71	68	101
i	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51
$P_{128}(i)$	6	39	36	69	102	7	8	41	74	107	104	9	42	75	72	105	10	43	40	73	106	11	12	45	78	111
i	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77
$P_{128}(i)$	108	13	46	79	76	109	14	47	44	77	110	15	16	49	82	115	112	17	50	83	80	113	18	51	48	81
i	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	100	101	102	103
$P_{128}(i)$	114	19	20	53	86	119	116	21	54	87	84	117	22	55	52	85	118	23	24	57	90	123	120	25	58	91
i	104	105	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120	121	122	123	124	125	126	127		
$P_{128}(i)$	88	121	26	59	56	89	122	27	28	61	94	127	124	29	62	95	92	125	30	63	60	93	126	31		

Table 1: DEFAULT permutation

Implementation

Table 8.2: `DEFAULT` test vectors (full cipher with 80 rounds)

1	Key	00000000000000000000000000000000
	Plaintext	00000000000000000000000000000000
	Ciphertext	02ec558a8f65dc7e53b326a1de9ade51
2	Key	ffffffffffffffffffffffffffffffff
	Plaintext	ffffffffffffffffffffffffffffffff
	Ciphertext	c823fe57c2e8b7b91db62aed3ad05e32
3	Key	98989898989898989898989898989898
	Plaintext	405b405b405b405b405b405b405b405b
	Ciphertext	cf18f09cd2d8f68ce64c3380272be64c
4	Key	329abdeb01339ab04465021f85417c3d
	Plaintext	0cebfdcbbcb01434d7111525143b94d1
	Ciphertext	7d1a340975be69a10a7d11d168b25b63

감사합니다