2019 암호경진대회

4번 문제 : 압축함수 충돌쌍 찾기

가능한 가장 큰 j 값에 대하여 아래 조건을 만족하는 $(Message, Message', j)$을 제시하고, 풀이 과정을 자세히 기술하시오.

1. $Message \neq Message'$
2. $Comp(j, Message) = Comp(j, Message')$

압축함수 $Comp$와 관련된 내용은 다음과 같음

○ 압축함수: $H = Comp(j, Message)$

- 입력 $Message$: 256비트 메시지 = $(m[0], m[1], ..., m[31])$ 크기 32의 8비트 단위벡터로 표현
- 출력 H : 128비트 값 = $(H[0], H[1], ..., H[15])$ 크기 16의 8비트 단위벡터로 표현
- $H = Comp(j, Message)$ 계산법
  1. $X_0 = (m[0], m[1], ..., m[15])$
  2. $K_0 = (m[16], m[17], ..., m[31])$
  3. **For** $i = 0, 1, ..., j-2$ **Do**:
  4. $\quad rk_i = (K_i[0], K_i[1], K_i[2], K_i[3])$
  5. $\quad X_{i+1} = Round(rk_i, X_i)$
  6. $\quad K_{i+1} = Round((i, i+1, i+2, i+3), K_i)$
  7. **Endfor**
  8. $rk_{j-1} = (K_{j-1}[0], K_{j-1}[1], K_{j-1}[2], K_{j-1}[3])$
  9. $X_j = Round'(rk_{j-1}, X_{j-1})$
  10. $H = X_j \oplus X_0 \oplus K_0$ (즉, $X_j[0] \oplus X_0[0] \oplus K_0[0], ..., X_j[15] \oplus X_0[15] \oplus K_0[15]$)

○ S-box: $y = S(x)$

- 8비트 값을 입력 받아 8비트 값을 출력하는 함수
- S-box는 AES의 S-box와 동일하며, (그림 1)의 입출력 표에 따라 계산됨
- 입력: $x = x_1 x_2 = x_1 0 + 0 x_2$
- 출력: $y = y_1 y_2$

ex) 입력이 $x = a8$ 이면, $x_1 = a, x_2 = 8$이므로 S-box 입출력 표에 의해 $y = S(a8) = c2$ 이다.

	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
10	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
20	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
30	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
40	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
50	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
60	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
70	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
80	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
90	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
a0	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
b0	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
c0	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
d0	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
e0	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
f0	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

(그림 1) S-box 입출력 표

○ 8비트 유한체 곱: $y = 2 \cdot x, y = 3 \cdot x$

- x = 8비트값

  x의 이진수 표현: $x = x_7x_6x_5x_4x_3x_2x_1x_0$ (x_7: x의 최상위 비트, x_0: x의 최하위 비트)

- $y = 2 \cdot x$ 계산법

  $x_7 = 0$ 일 때 → $y = x_6x_5x_4x_3x_2x_1x_0 0$

  $x_7 = 1$ 일 때 → $y = x_6x_5x_4x_3x_2x_1x_0 0 \oplus 00011011$ ($\oplus$: 비트별 배타적 논리합 XOR)

- $y = 3 \cdot x$ 계산법

  $y = 3 \cdot x = 2 \cdot x \oplus x$

ex1) $x = a8$이면, x의 이진수 표현은 $x = 10101000$

$2 \cdot x = 01010000 \oplus 00011011 = 01001011$

$3 \cdot x = 2 \cdot x \oplus x = 01001011 \oplus 10101000 = 11100011$

ex2) $x = 7b$이면, x의 이진수 표현은 $x = 01111011$

$2 \cdot x = 11110110$

$3 \cdot x = 2 \cdot x \oplus x = 11110110 \oplus 01111011 = 10001101$

○ 32비트 행렬곱: $y^T = M \bullet x^T$

- 32비트 값 x를 8비트 단위 벡터(크기 4인 1×4 행렬)로 표현: $x = (x[0], x[1], x[2], x[3])$
- 32비트 행렬 곱 $y^T = M \bullet x^T$ 계산법 (AES MixColumn 연산과 동일함)

$$y^T = \begin{bmatrix} y[0] \\ y[1] \\ y[2] \\ y[3] \end{bmatrix} = M \bullet x^T = \begin{bmatrix} 2\,3\,1\,1 \\ 1\,2\,3\,1 \\ 1\,1\,2\,3 \\ 3\,1\,1\,2 \end{bmatrix} \begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix} = \begin{bmatrix} 2 \bullet x[0] \oplus 3 \bullet x[1] \oplus x[2] \oplus x[3] \\ x[0] \oplus 2 \bullet x[1] \oplus 3 \bullet x[2] \oplus x[3] \\ x[0] \oplus x[1] \oplus 2 \bullet x[2] \oplus 3 \bullet x[3] \\ 3 \bullet x[0] \oplus x[1] \oplus x[2] \oplus 2 \bullet x[3] \end{bmatrix}$$

○ F 함수: $y = F(rk, x)$

- F 함수의 첫 번째 32비트 입력 $rk = (rk[0], rk[1], rk[2], rk[3])$와 두 번째 32비트 입력 $x = (x[0], x[1], x[2], x[3])$
- $y = F(rk, x)$ 계산법
  1. $a = x \oplus rk$ (즉, $a[i] = x[i] \oplus rk[i]$ for $i = 0,1,2,3$)
  2. $b = S(a)$ (즉, $b[i] = S(a[i])$ for $i = 0,1,2,3$)
  3. $y^T = M \bullet b^T$ (즉, $y^T = M \bullet (b[0], b[1], b[2], b[3])^T$)

○ 라운드 함수: $Y = Round(rk, X)$

- 라운드 함수 $Round$의 첫 번째 32비트 입력 $rk = (rk[0], rk[1], rk[2], rk[3])$, 두 번째 128비트 입력 X
  X의 8비트 단위 벡터 표현 $X = (X[0], X[1], ..., X[15])$
- $Y = Round(rk, X)$ 계산법
  1. $(Y[0], ..., Y[3]) = F(rk, (X[0], ..., X[3])) \oplus (X[4], ..., X[7])$
  2. $(Y[4], ..., Y[7]) = (X[8], ..., X[11])$
  3. $(Y[8], ..., Y[11]) = (X[12], ..., X[15])$
  4. $(Y[12], ..., Y[15]) = (X[0], ..., X[3])$

○ 라운드 함수: $Y = Round'(rk, X)$

- 라운드 함수 $Round'$의 첫 번째 32비트 입력 $rk = (rk[0], rk[1], rk[2], rk[3])$, 두 번째 128비트 입력 X
  X의 8비트 단위 벡터 표현 $X = (X[0], X[1], ..., X[15])$
- $Y = Round'(rk, X)$ 계산법
  1. $(Y[0], ..., Y[3]) = (X[0], ..., X[3])$
  2. $(Y[4], ..., Y[7]) = F(rk, (X[0], ..., X[3])) \oplus (X[4], ..., X[7])$
  3. $(Y[8], ..., Y[11]) = (X[8], ..., X[11])$
  4. $(Y[12], ..., Y[15]) = (X[12], ..., X[15])$

○ 테스트 벡터(16진수 표현)

$Message = m[0]\,m[1]\,m[2]\cdots m[30]\,m[31]$ = 00 01 02 03 04 05 06 07 08 09 0a 0b 0c 0d 0e 0f 0a 1a 2a 3a 4a 5a 6a 7a 8a 9a aa ba ca da ea fa

j	$H[0]\,H[1]\cdots H[15]$
1	0a 1a 2a 3a 48 36 fc 6c 8a 9a aa ba ca da ea fa
2	0c 72 b8 28 fd f3 3c 7d 8e 9e ae be c6 d6 e6 f6
3	b9 b7 78 39 09 06 c7 f3 82 92 a2 b2 c0 be 74 e4
4	4d 42 83 b7 c9 5a c4 25 84 fa 30 a0 75 7b b4 f5
5	8d 1e 80 61 b8 9b a5 6c 31 3f f0 b1 81 8e 4f 7b
6	fc df e1 28 5b f7 95 a4 c5 ca 0b 3f 41 d2 4c ad
7	1f b3 d1 e0 11 3b 4f 17 05 96 08 e9 30 13 2d e4
8	55 7f 0b 53 df a3 d4 f6 74 57 69 a0 d3 7f 1d 2c
9	9b e7 90 b2 cd 6f c4 22 97 3b 59 68 99 b3 c7 9f
10	89 2b 80 66 d2 47 6c 56 dd f7 83 db 57 2b 5c 7e