2020 암호분석경진대회 답안제출

2020. 08. 31

참가자 1	**성 명**	이창원
	소속	서울시립대학교 수학과
	휴대폰	
	E-mail	

참가자 2	**성 명**	장호빈
	소속	서울시립대학교 수학과
	휴대폰	
	E-mail	

참가자 3	**성 명**	김인성
	소속	서울시립대학교 수학과
	휴대폰	
	E-mail	

5번 문제 답안

정답:

23라운드

풀이:

이 문제에 사용된 해시함수의 라운드함수는 AES의 S-box와 Mixcolumns, xor 연산을 사용하여 설계되었다. 이 함수들은 모두 역함수가 있는 함수들이므로 일대일대응인 함수들이다. 그리고 16라운드까지 사용되는 W0~W15는 모두 독립적이다. 이를 사용하여 16라운드까지는 매우 쉽게 임의의 Y값에 대해 Y를 출력 값으로 갖는 CF의 어떤 입력 값 X를 찾을 수 있다.

8라운드 해시함수에서 A_0~H_0가 고정되면 W_0~W_7에 의해 해시 값이 결정되며 일대일 대응이므로 2^{256}개의 임의의 해시 값을 만족하는 W_0~W_7이 존재하므로 A_0~H_0를 0으로 고정하므로서 해시함수의 라운드 함수를 역으로 계산하여 W_0~W_7을 구할 수 있다.

16라운드 해시함수 또한 A_0~H_0, A_8~H_8을 0으로 고정하게 되면 8라운드와 마찬가지로 라운드함수를 역으로 계산하여 W_0~W_{15}를 구할 수 있기 때문에 임의의 해시 값에 대해 16라운드까지 2^{nd} preimage 값을 구할 수 있다.

16라운드 이하의 2^{nd} preimage 값을 구하는 의사코드(pseudo code)는 다음과 같다.

L<=8라운드
1. 주어진 값 Y = { Y_0, …, Y_7 }
2. 입력 값 : (A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0) with { 0x00, 0x00, 0x00, 0x00 } 32-bit
3.
for i from 0 to L-1
    T = { $Y_{7-i}[0]$, $Y_{7-i}[1]$, $Y_{7-i}[2]$, $Y_{7-i}[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus G_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus G_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus G_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus G_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus F_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus F_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus F_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus F_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus E_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus E_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus E_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus E_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus D_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus D_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus D_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus D_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus C_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus C_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus C_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus C_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus B_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus B_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus B_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus B_i[3]$ }
    T = { $S^{-1}(MDS^{-1}(T[0]))\oplus A_i[0]$, $S^{-1}(MDS^{-1}(T[1]))\oplus A_i[1]$, $S^{-1}(MDS^{-1}(T[2]))\oplus A_i[2]$, $S^{-1}(MDS^{-1}(T[3]))\oplus A_i[3]$ }
    W_i={ T[0], T[1], T[2], T[3] }
    (A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1}) = Round(A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, W_i)
//Round 함수는 CF함수에서 사용되는 Round 함수와 동일
if i>L
    W_i = { 0x00, 0x00, 0x00, 0x00 }   //라운드보다 큰 인덱스를 갖는 W 값은 의미가 없음
4. (A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0, W_0, W_1, W_2, W_3, W_4, W_5, W_6, W_7)을 출력한다.

5번 문제 답안

8<L<=16라운드

1. 주어진 값 $Y = \{ Y_0, \cdots, Y_7 \}$

2. 입력 값 : $(A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0)$ with { 0x00, 0x00, 0x00, 0x00 } 32-bit
$(A_8, B_8, C_8, D_8, E_8, F_8, G_8, H_8)$ with { 0x00, 0x00, 0x00, 0x00 } 32-bit

3.

for i from 0 to 7

$T = \{ 0x00, 0x00, 0x00, 0x00 \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus G_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus G_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus G_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus G_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus F_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus F_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus F_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus F_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus E_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus E_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus E_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus E_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus D_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus D_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus D_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus D_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus C_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus C_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus C_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus C_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus B_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus B_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus B_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus B_i[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus A_i[0], S^{-1}(MDS^{-1}(T[1]))\oplus A_i[1], S^{-1}(MDS^{-1}(T[2]))\oplus A_i[2], S^{-1}(MDS^{-1}(T[3]))\oplus A_i[3] \}$

$W_i = \{ T[0], T[1], T[2], T[3] \}$

$(A_{i+1}, B_{i+1}, C_{i+1}, D_{i+1}, E_{i+1}, F_{i+1}, G_{i+1}, H_{i+1}) = Round(A_i, B_i, C_i, D_i, E_i, F_i, G_i, H_i, W_i)$

for i from 0 to L

$T = \{ Y_{7-i}[0], Y_{7-i}[1], Y_{7-i}[2], Y_{7-i}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus G_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus G_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus G_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus G_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus F_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus F_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus F_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus F_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus E_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus E_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus E_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus E_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus D_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus D_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus D_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus D_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus C_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus C_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus C_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus C_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus B_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus B_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus B_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus B_{i+8}[3] \}$

$T = \{ S^{-1}(MDS^{-1}(T[0]))\oplus A_{i+8}[0], S^{-1}(MDS^{-1}(T[1]))\oplus A_{i+8}[1], S^{-1}(MDS^{-1}(T[2]))\oplus A_{i+8}[2], S^{-1}(MDS^{-1}(T[3]))\oplus A_{i+8}[3] \}$

$W_{i+8} = \{ T[0], T[1], T[2], T[3] \}$

$(A_{i+9}, B_{i+9}, C_{i+9}, D_{i+9}, E_{i+9}, F_{i+9}, G_{i+9}, H_{i+9}) = Round(A_{i+8}, B_{i+8}, C_{i+8}, D_{i+8}, E_{i+8}, F_{i+8}, G_{i+8}, H_{i+8}, W_{i+8})$

//Round 함수는 CF함수에서 사용되는 Round 함수와 동일

if i>L

$W_i = \{ 0x00, 0x00, 0x00, 0x00 \}$ //라운드보다 큰 인덱스를 갖는 W 값은 의미가 없음

4. $(A_0, B_0, C_0, D_0, E_0, F_0, G_0, H_0, W_0, W_1, W_2, W_3, W_4, W_5, W_6, W_7)$을 출력한다.

5번 문제 답안

그러나 17라운드 이상부터 사용되는 W_i는 앞의 W_0~W_{16}에 의해 결정되게 되어 위와 같은 방법을 사용할 때 바로 구하는 것이 어렵다. L 라운드(16<L<24)라고 할 때 W_0~W_{15}를 임의의 값으로 선택하게 되면 W_{16}~W_{L-1}이 계산이 되고 23라운드 이하이므로 A_{16}~A_{L-7}이 A_L~H_L 중 뒤에서부터 24-L개 만큼과 동일한 값을 가지게 된다. 그러면 A_0~H_0에 임의의 값을 넣어 해시함수를 계산할 때 16라운드만 계산하여 A_{16}~H_{16}을 구하고 A_L~H_L과 W_{16}~W_{L-1}을 통해 A_{16}~H_{16}을 계산하여 둘을 비교할 때 일치할 확률은 $2^{32(L-16)}$이므로 23라운드까지는 2^{253}이내의 계산력으로 2^{nd} preimage 값을 찾아낼 수 있을 것으로 기대할 수 있다. 이에 대한 의사 코드는 다음과 같다.

Preparation
1. 주어진 값 Y = { Y_0, …, Y_7 }
2. W_0~W_{15}를 임의의 값으로 선택
3. W_{16}~W_{L-1}을 계산
First phase
1. A_0~H_0를 임의로 선택
2. A_0~H_0와 W_0~W_{15}를 통해 A_{16}~A_{L-7}을 계산
3. 결과 값을 table에 저장
4. 이 과정을 $2^{32(L-16)}$번 시행
Second phase
1. First phase에서 선택된 A_0~H_0를 통해 A_L~H_L을 계산
2. A_L~H_L과 W_{16}~W_{L-1}을 통해 A_{16}~H_{16}을 계산
3. A_{16}~H_{16} 중 뒤에서부터 24-L개 만큼을 First phase에서 table에 저장된 값과 비교
4. 일치하는 값이 나오면 종료 후 A_0~H_0와 W_0~W_{15} 출력

이 때 A_L~H_L과 W_{16}~W_{L-1}을 통해 A_{16}~H_{16}을 계산하는 연산은 아래와 같다.
for i=L to i=17
    T = Round(B_i, …, H_i, A_i, W_{i-1})
(A_{16}, …, H_{16}) = (T[0], …, T[7])

그리고 CF함수와 SHA-2는 구조적으로 흡사하다. 그래서 Preimages for Step-Reduced SHA-2 논문을 참고해보자. 이 논문에서는 SHA-2의 Message Expansion 과정에서 각각 다른 두 메시지에 독립적인 두 개의 chunk를 만들고 이를 이용하여 MITM공격을 하게 된다. 그리고 SHA-2의 특성을 이용한 Message stealing과 Message compensation 기법을 사용하여 두 chunk의 길이의 합을 33으로 늘려 33step에 MITM 공격을 적용할 수 있도록 하고 inderect partial matching 기법을 사용하여 9step을 더 늘려 총 42step에 대한 MITM 공격을 진행하게 된다. 논문에서 나온 내용 중 CF함수에 적용할 수 있는 것은 독립적인 두 개의 chunk로 나누는 것과 matching point에서 birthday attack를 통해 확률을 감소시키는 것이라고 생각이 된다. 그래서 논문과 같은 기법을 사용하여 CF함수의 Message expansion을 이용하여 독립적인 두 개의 chunk로 나눈 결과 최대 두 chunk의 길이의 합이 25가 되었다. 그래서 CF함수의 경우 25라운드 정도를 논문에 나온 기법을 사용하여 2^{253} 이내의 계산량으로 2^{nd} preimage값을 얻을 수 있을 것이라고 생각한다.