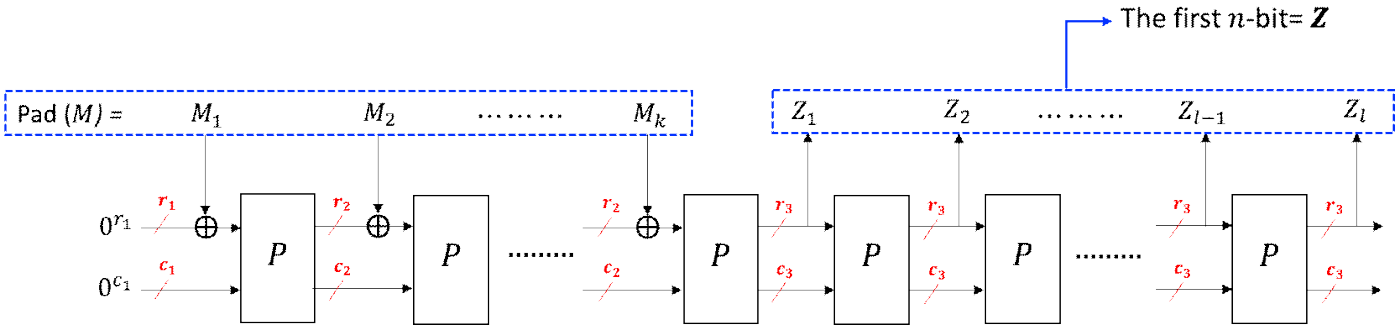
2022 암호분석경진대회

6번 문제

아래의 그림과 같이 스펀지 기반 해쉬함수를 설계하고자 한다. 입력 값은 임의의 비트 길이 (이때, 비트 길이가 0인 것을 포함)의 메시지 M이고, 출력값 Z는 $Z_1\|Z_2\|\cdots\|Z_l$의 처음 n 비트 값으로 정의한다. (이때, 각각의 Z_i 값들은 r_3 비트임.)



<스펀지 기반 해쉬함수>

P는 NIST의 LWC 프로젝트 최종 라운드에 올라간 해쉬함수인 Ascon-Hash에 사용된 320 비트 치환함수 (12 라운드로 구성됨.) 를 뜻한다. 0^x는 비트 0이 x번 연접한 값을 뜻한다. 가령, 0^2은 2 비트 값 00을 뜻한다. r_1,r_2,r_3,c_1,c_2,c_3 는 비트 길이를 뜻하며, $r_1+c_1=r_2+c_2=r_3+c_3=320$ 의 관계를 갖는다. 임의의 비트 길이 (0 포함)의 메시지 M이 주어져 있을 때, 패딩 방법 Pad를 이용하여 $\text{Pad}(M)=M_1\|M_2\|\cdots\|M_k$를 생성한다. 이때, k가 2 이상인 경우, 각각의 M_i $(M_i\neq 1)$ 값들은 r_2 비트이고, M_1은 r_1 비트이다.

위의 그림의 해쉬 함수 동작 과정을 순차적으로 설명을 하면 다음과 같다. 처음 초기치는 0^{320}이 되고, 상태 값의 처음 r_1 비트와 M_1에 $\oplus$ (배타적 논리합)을 적용하여 상태 값을 업데이트하고, 업데이트된 상태 값에 치환함수 P를 적용하여 출력 상태 값을 생성한다. 이어서, 출력 상태 값의 처음 r_2 비트와 M_2에 $\oplus$ (배타적 논리합)을 적용하여 상태 값을 다시 업데이트하고, 업데이트된 상태 값에 치환함수 P를 적용하여 출력 상태 값을 생성한다. 이런 식으로 M_k까지 반복 하여 상태 값을 업데이트 한다. 그리고 나서, 상태 값의 처음 r_3 비트 값을 Z_1으로 정의하고, 상태 값을 다시 P로 업데이트 한 후, 상태 값의 처음 r_3 비트 값을 Z_2로 정의한다. 똑같은 방식으로 Z_l까지 정의한다.

[문제]

위에서 설계한 해쉬함수에서 아직 구체적으로 정하지 않은 정보들이 있다.

첫째, 패딩방법 Pad를 어떻게 정의할 것인가?

둘째, $r_1,r_2,r_3,c_1,c_2,c_3,n$ 값들을 어떻게 정의할 것인가?

이에 대하여, 제1역상 공격 (Preimage attack), 제2역상 공격 (Second-preimage attack), 충돌쌍 공격 (Collision attack) 각각의 공격에 대한 128 비트 안전성을 가지면서 (즉, 각각의 공격 복잡도가 최소 2^{128} 연산을 요구함. 이때 연산의 기본 단위는 P 함수 한번 수행하는 것을 1로 함), 구현 속도 측면에서 최대한 효율적이 되도록, 패딩방법 Pad와 $r_1,r_2,r_3,c_1,c_2,c_3,n$ 값들을 정의하고, 그 이유를 논리적으로 설명하여라.