해시함수 제안 논문 리뷰

IT융합공학부 권혁동

한성대학교 HANSUNG UNIVERSITY

CryptoCraft LAB
https://crypto.modoo.at

Contents

CryptoCraft LAB

해시함수란?

- 데이터 검색을 빠르게 하기 위해 제안
- 특유의 성질을 활용하여 암호 분야에 활용

1. 입력 값에 따라 특정 고정 길이로 반환
2. 입력 값이 하나라도 달라지면 완전 다른 결과

해시함수란?

- 제 1 역상저항성
- 제 2 역상저항성
- 충돌저항성
  - 무작위 두개의 입력 값에서 같은 결과를 획득
  - 비둘기집 원리
- 눈사태효과
  - 적은 입력 값의 변화로 결과값의 큰 변화

제안 해시함수

- 입력 규격: 제한 없음
- 출력 규격: 288bits
- 라운드: 64
- 고속, 강한 눈사태 효과, 강한 충돌저항성

제안 해시함수

- 전처리 1단계: 패딩

입력 값이 512비트의 배수가 되도록 패딩 붙이기

L: 원본 입력 길이

z: 임의 길이의 패딩 길이, 모든 값은 0

1: 1비트 길이의 패딩, 값은 1

$$L + z + 1 = 448 \; mod \; 512$$

제안 해시함수

- 전처리 2단계: N-512bit 블록으로 나누기

각 512비트의 블록을 16개의 블록으로 나눔.
하나의 블록은 32비트를 차지

$$16 * (32\text{-bit word}) : \ M_1^{(i)}, M_2^{(i)}, \dots\dots\dots, M_{15}^{(i)}.$$

제안 해시함수

- 전처리 3단계: 초기화 벡터 H_0 생성

초기화 벡터는 288비트 길이
첫 9개의 소수(prime)의 제곱근을 취함
제곱근 값의 소수(decimal) 부분을 사용
획득한 값은 32비트로 표현

제안 해시함수

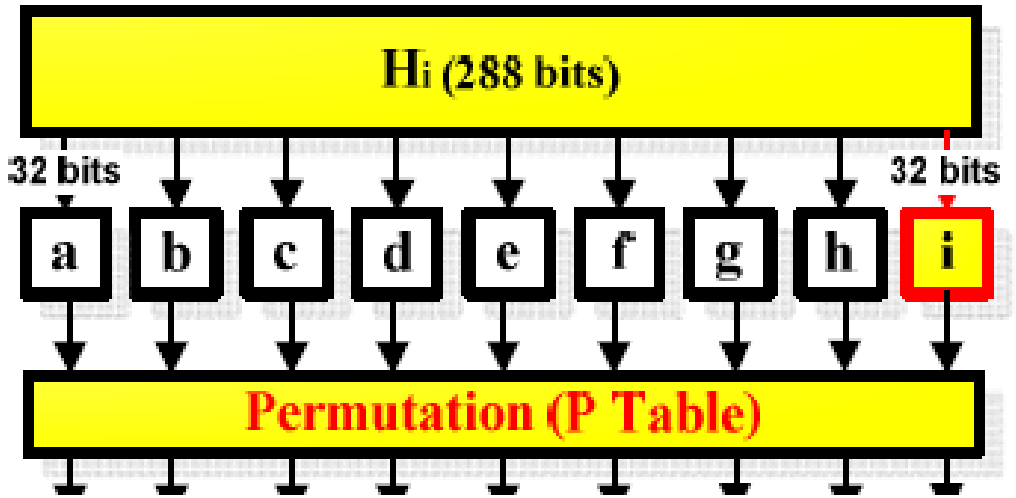
- 상수 정의: 연산에 사용하는 상수 정의

$$\boldsymbol{c}(x,y,z) = (x \wedge y) \oplus (\bar{x} \wedge z)$$
$$\boldsymbol{g}(x,y,z) = (x \wedge y) \oplus (x \wedge z) \oplus (y \wedge z)$$
$$\sum_0^{\{288\}}(x) = RR_{[2]}(x) \oplus RR_{[13]}(x) \oplus RR_{[22]}(x)$$
$$\sum_1^{\{288\}}(x) = RR_{[6]}(x) \oplus RR_{[11]}(x) \oplus RR_{[25]}(x)$$
$$\boldsymbol{\sigma}_0^{\{288\}}(x) = RR_{[7]}(x) \oplus RR_{[18]}(x) \oplus SR_{[3]}(x)$$
$$\boldsymbol{\sigma}_1^{\{288\}}(x) = RR_{[17]}(x) \oplus RR_{[19]}(x) \oplus SR_{[10]}(x)$$

제안 해시함수

- 해시 1단계: H_i를 9개의 레지스터에 분할

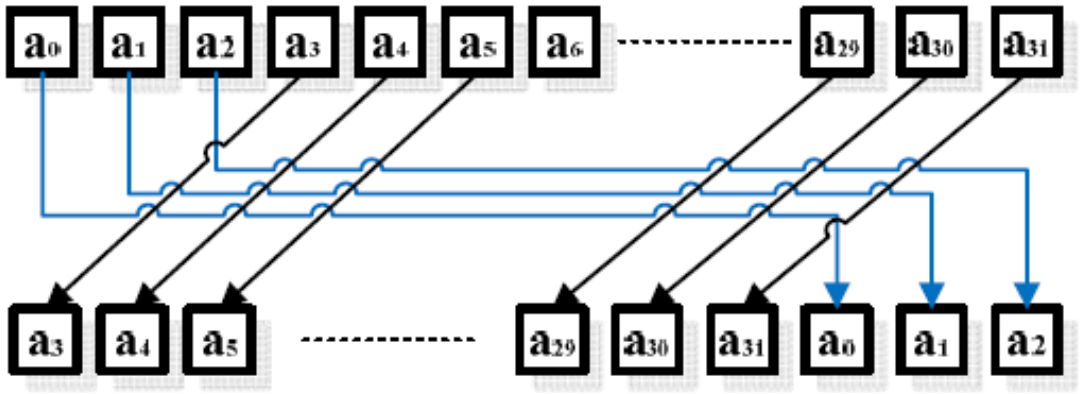
a ~ i로 칭하며 각각 32비트 길이를 차지



제안 해시함수

- 해시 1단계: P-table을 통해 비트를 섞음

각각 왼쪽 시프트 연산을 C_p번 행함
이때 C_p의 값은 소수이며 임의로 정의



제안 해시함수

- 해시 2단계: 라운드 함수

각 라운드마다 세가지의 입력 값 필요

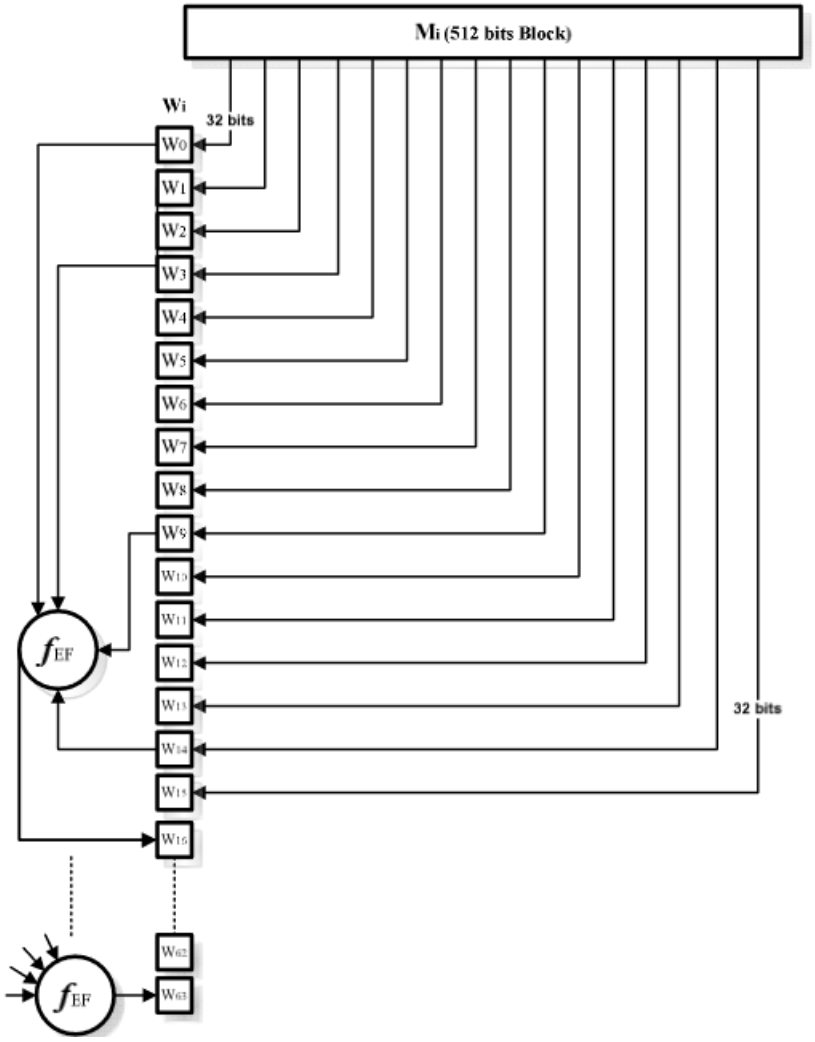
1. H: 직전 단계의 해시 연산 값, 288비트
2. W: 메시지에서 획득한 워드, 32비트
3. K: 라운드 상수, 32비트

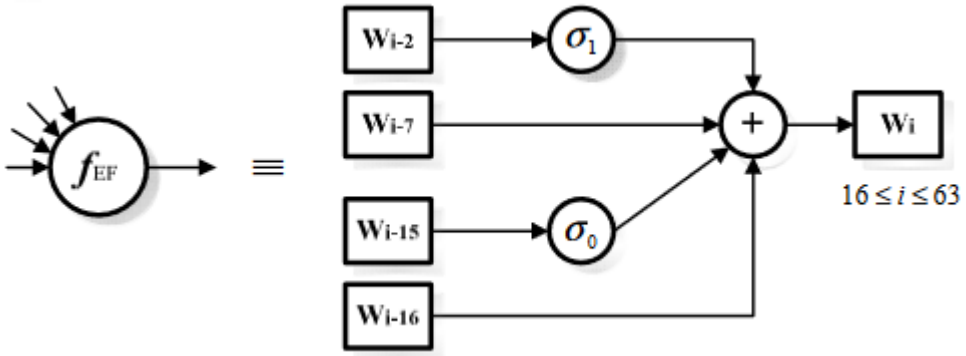
제안 해시함수

- 해시 2단계: 라운드 함수

$W_{0\sim15}$ -> 입력 메시지의 1~16번째 워드
$W_{16\sim63}$ -> 특수한 연산

$$\boldsymbol{\sigma}_1^{\{288\}}(W_{t-2}) + W_{t-7} + \boldsymbol{\sigma}_0^{\{288\}}(W_{t-15}) + W_{t-16}$$


Mi (512 bits Block)
Wi
32 bits
W0
W1
W2
W3
W4
W5
W6
W7
W8
W9
W10
W11
W12
W13
W14
W15
W16
W62
W63
fEF
fEF
32 bits

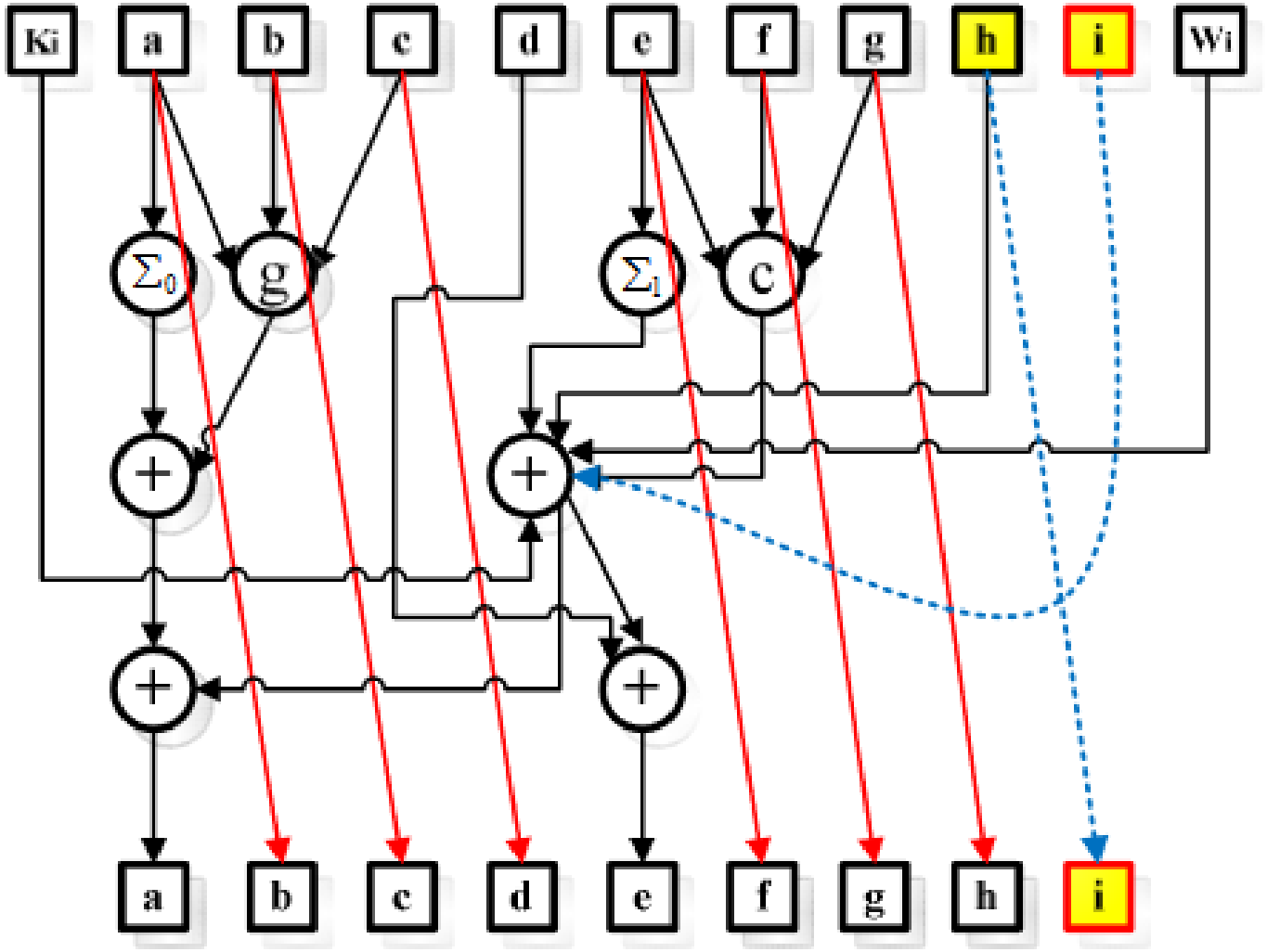

fEF
≡
Wi-2
Wi-7
Wi-15
Wi-16
σ1
σ0
+
Wi
16 ≤ i ≤ 63

제안 해시함수

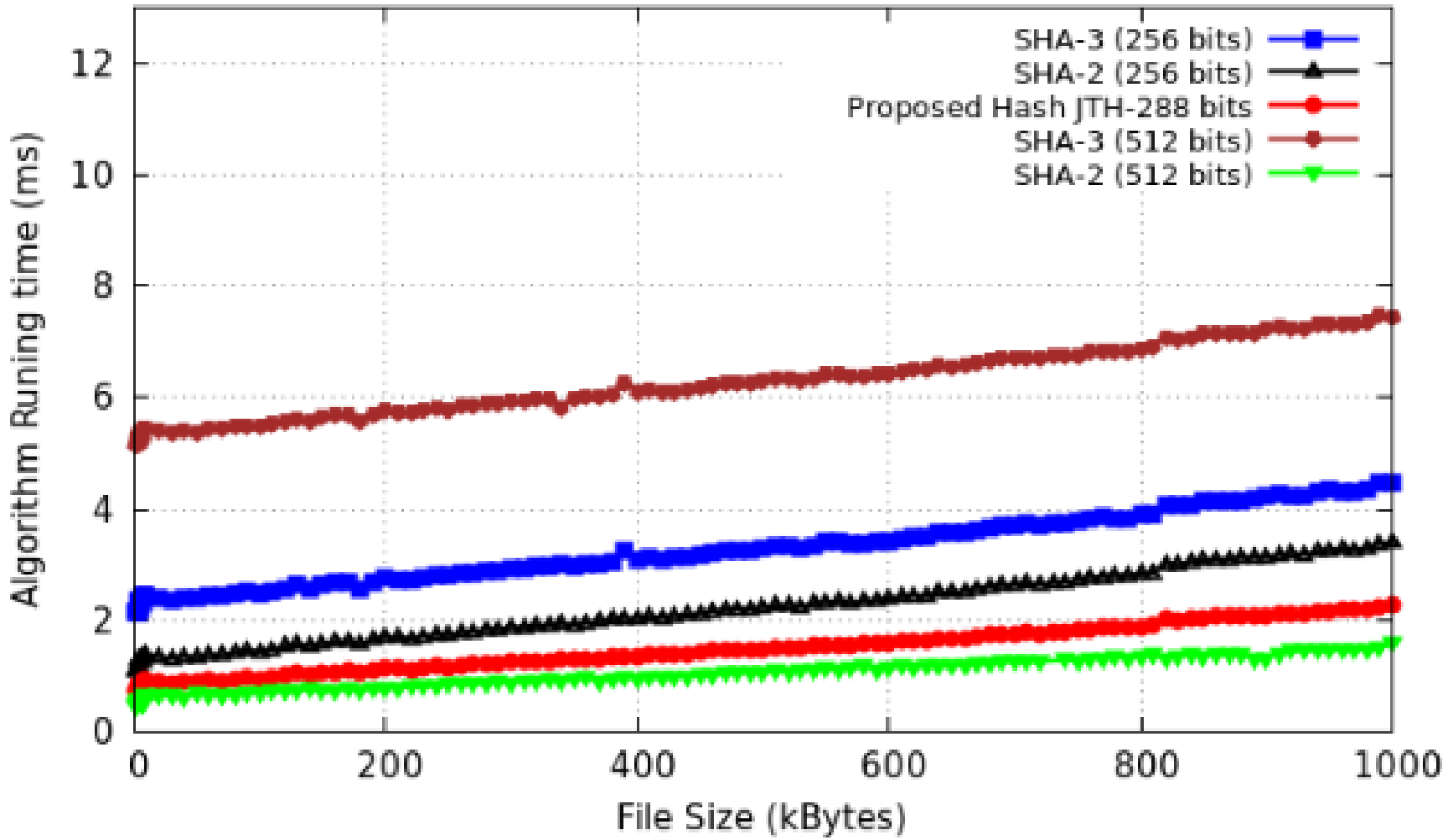
- 해시 2단계: 라운드 함수

라운드 상수
첫 64개의 소수(prime)의 세제곱근을 취함
세제곱근 값의 소수(decimal) 부분을 사용
각 소수는 32비트 길이로 64개를 형성

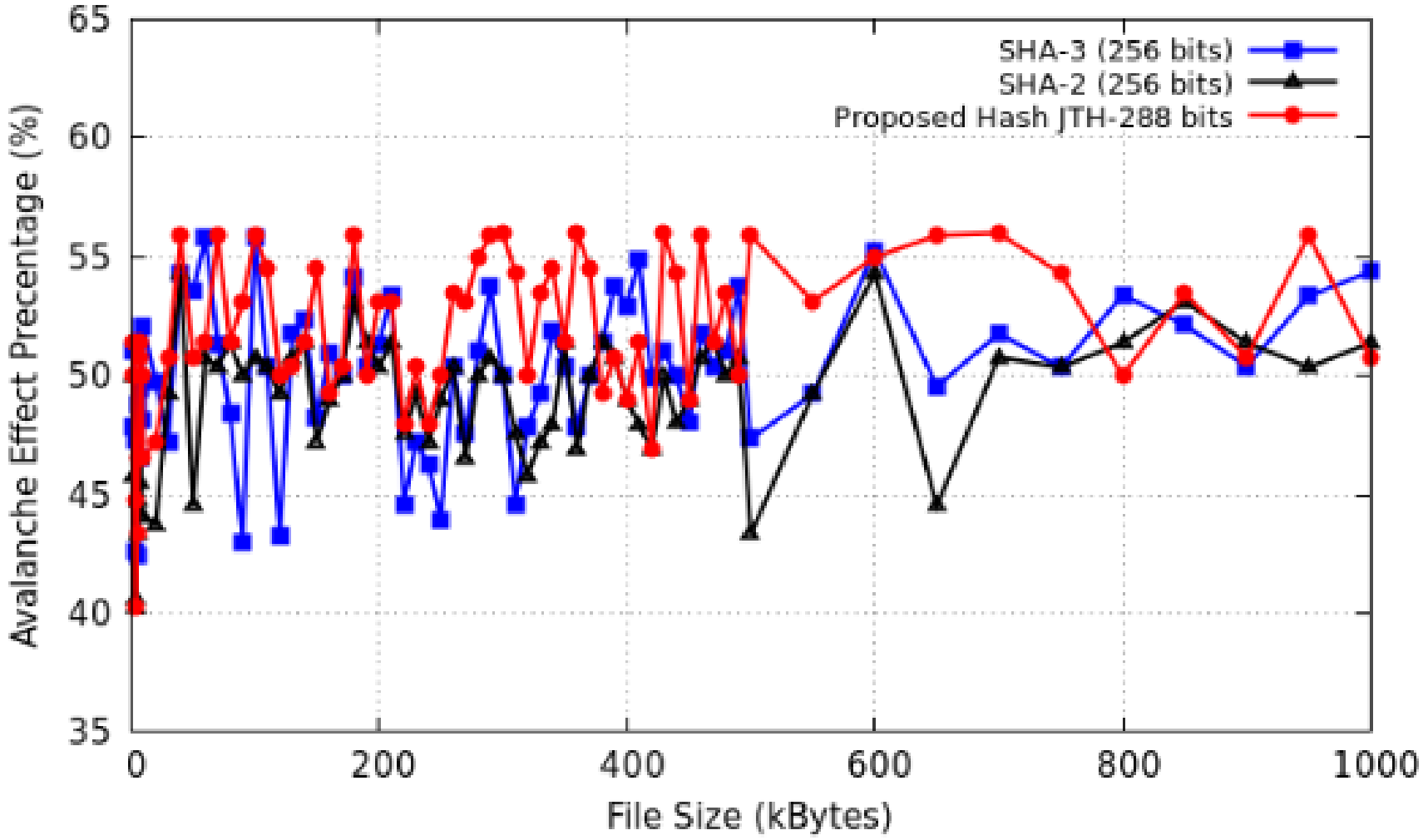
제안 해시함수



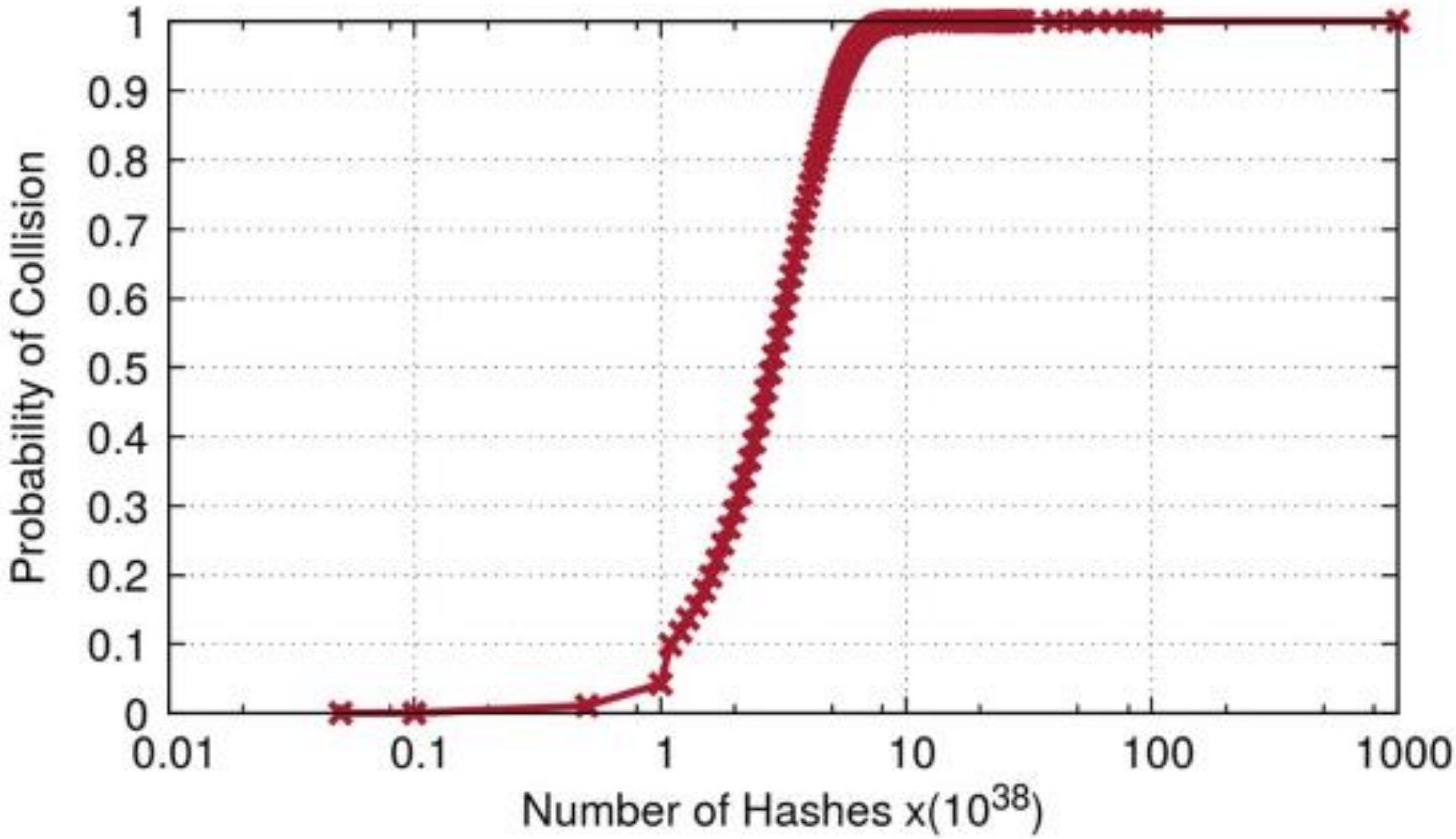
성능평가


SHA-3 (256 bits)
SHA-2 (256 bits)
Proposed Hash JTH-288 bits
SHA-3 (512 bits)
SHA-2 (512 bits)
Algorithm Runing time (ms)
0
2
4
6
8
10
12
0
200
400
600
800
1000
File Size (kBytes)

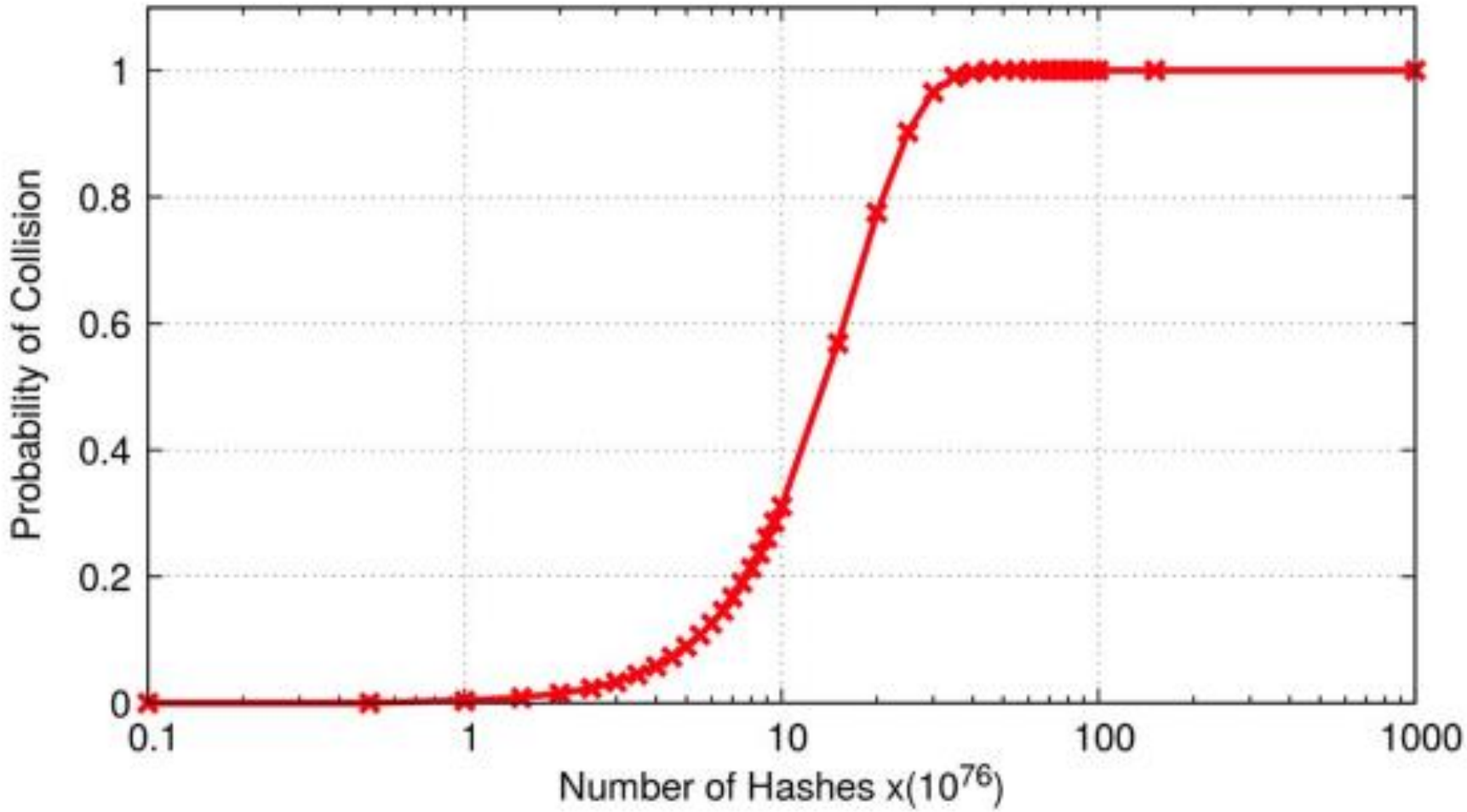
성능평가


SHA-3 (256 bits)
SHA-2 (256 bits)
Proposed Hash JTH-288 bits
Avalanche Effect Precentage (%)
File Size (kBytes)
65
60
55
50
45
40
35
0
200
400
600
800
1000

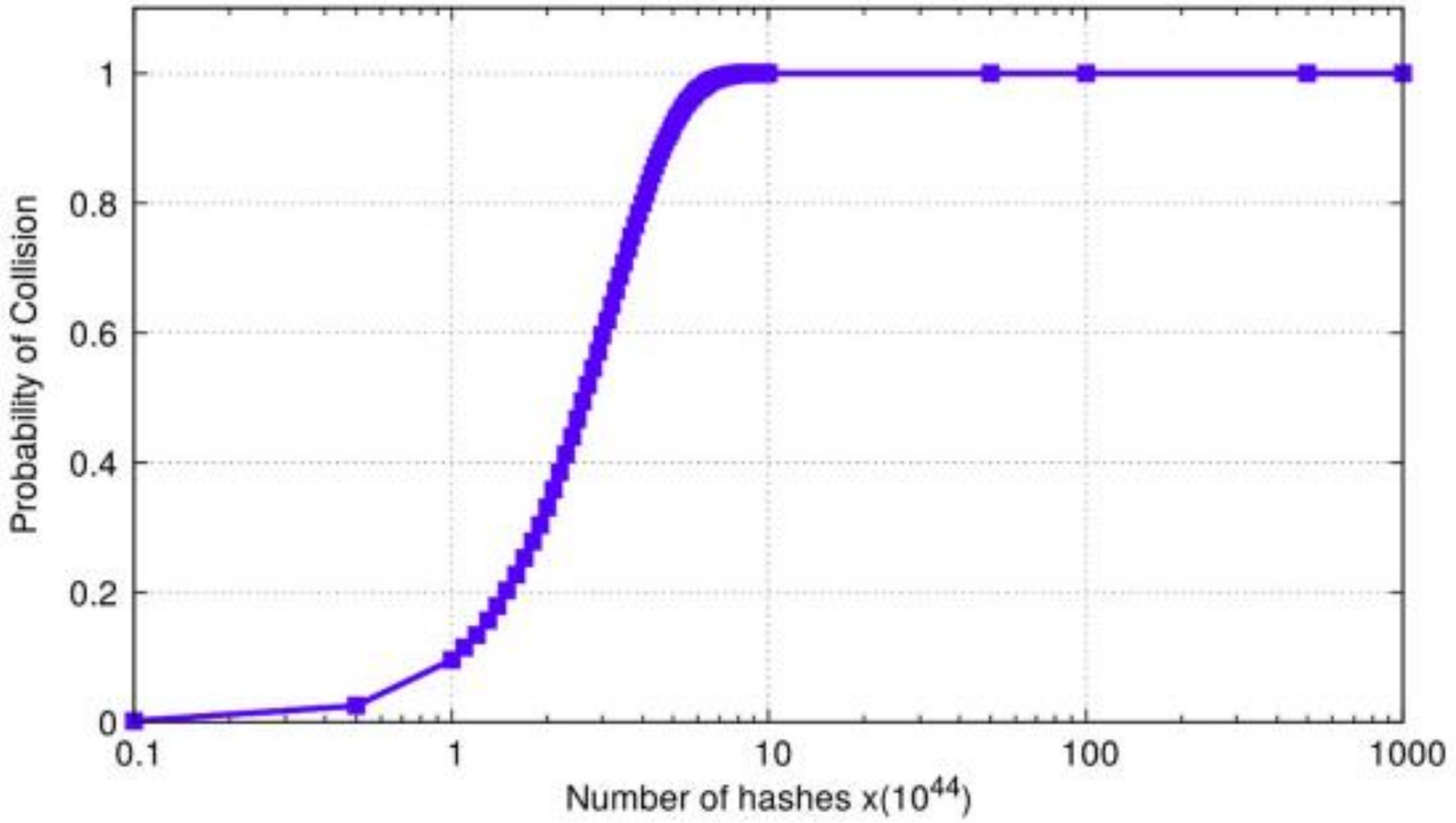
성능평가


Probability of Collision
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0.01
0.1
1
10
100
1000
Number of Hashes x(10^38)

성능평가


Probability of Collision
0
0.2
0.4
0.6
0.8
1
0.1
1
10
100
1000
Number of Hashes x(10^{76})

성능평가


Probability of Collision
0
0.2
0.4
0.6
0.8
1
0.1
1
10
100
1000
Number of hashes x(10^44)

결론

- 제안하는 해시는 CRN에서 사용할 수 있도록 제안
- 빠른 속도, 적정 수준의 보안성을 지님
- 다만 보안성이 검토된 부분은 아님
- 이 외에 두 가지 정도의 문제가 있음

결론

- P-table의 문제
- C_p 횟수 만큼 시프트 연산을 취함
- C_p 값이 지정된 것이 아니라 임의의 값
- 검증시에 서로 C_p 값을 교환해둬야 하는 문제 발생

결론

- 독자 출력 규격의 문제
- 표준 해시함수인 SHA는 224, 256, 384, 512 규격
- 제안 해시 함수는 288 규격
- 기존 규격에 맞지 않으므로 사용 가능성 낮음