通信情報保護學會學會誌
第 9 券 第 4 號, 1999. 12

Universal 해쉬 함수

박 상 우*, 지 성 택*, 박 춘 식*

요 약

본 논문은 유니버설 해쉬 함수에 대한 고찰로서, 유니버설 해쉬 함수의 정의 및 성질, 유니버설 해쉬 함수를 이용한 메시지 인증 방식 설계 방법인 Wegman-Carter 패러다임의 내용, 그리고, 제안된 각종 유니버설 해쉬 함수의 구성 및 특성을 내용으로 한다.

I. 서 론

메시지 인증(message authentication)은 정보 보호의 중요한 분야이며, 네트워크를 통하여 전송되는 정보가 많아질수록 안전하고, 고속이며, 소프트웨어 기반의 메시지 인증 기술이 더욱 필요하게 된다.

일반적인 메시지 인증 절차는 다음과 같다. 두 사용자 A와 B는 비밀키 a를 공유하고 있다. 메시지 인증 방식은 서명 알고리즘 S와 검증 알고리즘 V로 구성된다. A가 메시지 X를 B에게 송신하고자 하면, A는 B와 공유한 비밀키 a를 사용하여 메시지 X에 대한 MAC(Message Authentication Code)으로 $MAC = S_a(x)$를 생성하고, (X, MAC)을 B에게 전송한다. (X, MAC)을 수신한 B는 동일한 비밀키 a를 사용하여 $V_a(X, MAC) = 1$인지를 검증한다.

메시지 인증 방식의 안전성은 참고 문헌[1]에 정의되어 있다. 메시지 인증 방식에 대한 공격은 공격자가 메시지를 위장하는 위장 공격(forgery attack)으로, 공격자에게 S_a와 V_a에 대한 오라클(oracle)이 주어진 가정에서 공격자가 오라클에서 제공하지 않은 어떤 메시지 X에 대하여 $V_a(X, MAC) = 1$을 얻는다면, 공격자는 위장 공격에 성공하였다고 한다. 안전한 메시지 인증 방식이란 이러한 위장 공격이 계산상 불가능(computationally infeasible)한 것을 의미한다.

가장 널리 사용되는 메시지 인증 방식은 블록 알고리즘을 CBC(Cipher Block Chaining) 모드로 동작시키는 것으로 사용된 블록 알고리즘이 DES (Data Encryption Standard)[2]인 경우에 이에 대한 안전성이 분석된 바 있다[1]. 또 다른 방법으로는 MD5[3]와 같은 해쉬 함수를 사용하는 방법이다. 해쉬 함수를 사용하여 메시지 인증 방식을 설계하기 위한 많은 연구가 있었다.[4,5,6] 이들은 DES-CBC 방식에 의한 메시지 인증 방식보다는 속도 면에서 효율적이나 메시지 인증 방식의 안전성이 증명되지 않는 단점을 가진다.

메시지 인증의 문제는 초기에는 정보 이론 관점에서 연구되었으며[7], 최근에는 정보 보호 기술 측면에서 연구되고 있다[8,9,10]. Wegman과 Carter가 제안한 메시지 인증 방식 설계 방법은 특정 조건을 만족하는 해쉬 함수와 의사 랜덤 함수(pseudo random function)를 사용하는 것으로, 특정 조건을 만족하는 해쉬 함수를 유니버설 해쉬 함수(universal hash function)라 하며, 이들이 제안한 메시지 인증방식 설계 방법을 Wegman-Carter 패러다임이라 부른다[10]. Wegman-Carter 패러다임의 장점은 해쉬 함수 계산이 정보 보호 측면의 해쉬 함수에 비하여 수행 속도가 빠르며, 설계된 메시지 인증 방식의 안전성을 증명할 수 있다는 것이다.

본 논문은 유니버설 해쉬 함수와 이를 이용한 메시지 인증 방식 설계 방법인 Wegman-Carter 패러다임에 관한 것으로, 유니버설 해쉬 함수의 정의 및 성질, Wegman-Carter 패러다임의 내용, 그리고, 제안된 각종 유니버설 해쉬 함수의 구성 및 특성을 내용으로 한다.

* 한국전자통신연구원

II. 유니버설 해쉬 함수의 정의 및 성질

1. 유니버설 해쉬 함수

본 절에서는 참고 문헌[9,10]을 바탕으로 유니버설 해쉬 함수의 정의 및 몇 가지 성질을 기술한다.

정의 1. *해쉬 함수 패밀리(family of hash functions) H는 다음으로 정의된다.*

$$H=\{h: A=\{0,1\}^{*} \rightarrow B=\{0,1\}^{b}\}$$

즉, 해쉬 함수 패밀리는 공집합이 아닌 동일한 정의역(domain) A와 치역(range) B를 가지는 함수 h 들의 집합이다. 다음은 유니버설 해쉬 함수의 기본 정의들이다.

정의 2. *$H=\{h: A=\{0,1\}^{*} \rightarrow B=\{0,1\}^{b}\}$가 서로 다른 $x,x' \in A$에 대하여,*

$$\Pr_{h \in H}[h(x)=h(x')] \leq \varepsilon$$

인 경우, 해쉬 함수 패밀리 H 를 ε-almost universal이라 하며, $\varepsilon-AU$라 표기한다.

정의 3. *$H=\{h: A=\{0,1\}^{*} \rightarrow B=\{0,1\}^{b}\}$가 서로 다른 $x,x' \in A$와 모든 $c \in \{0,1\}^{b}$ 에 대하여,*

$$\Pr_{h \in H}[h(x) \oplus h(x')=c] \leq \varepsilon$$

인 경우, H를 ε-almost XOR universal이라 하며, ε-AXU라 표기한다.

$\varepsilon=\max_{x \neq x'} \Pr_{h}[h(x)=h(x')]$을 충돌 확률(collision probability)이라 하며, 유니버설 해쉬 함수 설계의 목표는 충돌 확률 ε을 가능한 한 작게 하며, 함수 계산 속도는 빠르게 하는 것이다. 다음은 유니버설 해쉬 함수의 성질들이다.

정리 1. *$H=\{h: \{0,1\}^{a} \rightarrow \{0,1\}^{b}\}$이라 하고, $H^{m}=\{h: \{0,1\}^{am} \rightarrow \{0,1\}^{bm}\}$은 각 원소들은 해쉬 함수 패밀리 H와 동일하나, $|x_i| = a$에 대해서, $h(x_1 x_2 \cdots x_m)=h(x_1)\|h(x_2)\|\cdots\|h(x_m)$이 되는 해쉬 함수 패밀리로 정의하자. 그러면, H가 $\varepsilon-AU$ 이면, H^{m}도 $\varepsilon-AU$ 이다.*

정리 2. *해쉬 함수 패밀리 $H_1=\{h: A \rightarrow \{0,1\}^{b_1}\}$과 $H_2=\{h: A \rightarrow \{0,1\}^{b_2}\}$에 대해서, 해쉬 함수 패밀리 $H_1 \& H_2=\{h: A \rightarrow \{0,1\}^{b_1+b_2}\}$ 을 $(h_1,h_2)(x) = h_1(x) \| h_2(x)$에 의해 정의하자. 그러면, H_1이 ε-AU 이고, H_2가 ε-AU 이면, $H_1 \& H_2$는 $\varepsilon_1 \varepsilon_2$-AU 이다.*

정리 3. *해쉬 함수 패밀리 $H_1=\{h: \{0,1\}^{a} \rightarrow \{0,1\}^{b}\}$과 $H_2=\{h: \{0,1\}^{b} \rightarrow \{0,1\}^{c}\}$에 대해서, 해쉬 함수 패밀리 $H_1 \circ H_2=\{h: \{0,1\}^{a} \rightarrow \{0,1\}^{c}\}$을 $(h_1,h_2)(x)=h_2(h_1(x))$으로 정의하자. 그러면, $H_1=\{h: A \rightarrow B\}$이 ε-AU 이고, $H_2=\{h: B \rightarrow C\}$가 ε-AXU 이면, $H_1 \circ H_2=\{h: A \rightarrow C\}$는 $(\varepsilon_1+\varepsilon_2)$-AXU 이다.*

2. Wegman-Carter 패러다임

본 절에서는 유니버설 해쉬 함수와 의사 랜덤 함수를 사용하여 안전성이 증명 가능한 메시지 인증 방식을 설계하는 Wegman-Carter 패러다임을 소개한다[7].

Wegman-Carter 패러다임은 해쉬 함수 패밀리 $H=\{h: A \rightarrow \{0,1\}^{b}\}$를 이용하여 메시지 인증 방식을 설계하는 방법으로 전체 구성에 의해 Universal-hash-and-then-encrypt 방식으로 모델화 된다. 서명자 S와 검증자 V는 유니버설 해쉬 함수 $h \in H$와 무한 랜덤 스트링(infinite random string) $P=P_1 P_2 P_3 \cdots, |P_i|=b$를 공유한다. 그러면, (h,P)는 서명자와 검증자가 공유한 키이다. 한편, 서명자는 0으로 초기화된 counter(cnt)를 관리한다. 그러면, 서명자의 메시지 x에 대한 MAC 생성 방법은 다음과 같다.

- 단계 1 : 서명자는 cnt를 증가한다.
- 단계 2 : MAC $\sigma=(cnt, P_{cnt} \oplus h(x))$를 계산한다.
- 단계 3 : σ와 메시지 x를 검증자에게 전송한다.

검증자의 검증 방법은 다음과 같다.

- 서명자로부터 수신한 MAC (i,s)에 대하여, s

$= Pi \oplus h(x)$ 임을 확인한다.

Wegman-Carter 패러다임은

- 유니버설 해쉬 함수에 의한 해쉬 과정
- 무한 랜덤 스트링에 의한 암호화 과정

으로 구성된다. Wegman-Carter 패러다임으로 설계된 메시지 인증 방식의 위장 공격에 대한 안전성은 유니버설 해쉬 함수의 충돌 확률에 의존한다.

정리 4. *H를 ε-AXU라 하자. 공격자가 Wegman-Carter paradigm으로 구성된 MAC을 위장하는데 성공할 확률 δ는 δ≤ε 이다.*

Wegman-Carter 패러다임에서 무한 랜덤 스트링 P를 사용하는 것은 현실적으로 매우 어렵기 때문에, 무한 랜덤 스트링 P 대신에 랜덤 인덱스 (random index) $a \in \{0,1\}^K$와 의사 랜덤 함수 $F:\{0,1\}^K \times \{0,1\}^l \rightarrow \{0,1\}^b$를 사용하는 방법이 있다. 서명자는 초기에 0 인 counter $cnt \in \{0,1\}^l$을 관리하며, 서명자와 검증자는 랜덤한 $a \in \{0,1\}^K$와 $h \in H$를 공유한다. 그러면, 서명자의 메시지 x에 대한 MAC 생성 방법은 다음과 같다.

- 단계 1 : 서명자는 cnt를 증가한다.
- 단계 2 : $\sigma = (cnt, F_a(cnt) \oplus h(x))$를 계산한다.
- 단계 3 : σ와 메시지 x를 검증자에게 전송한다.

검증자의 검증 방법으로 서명자로부터 수신한 MAC (i,s)에 대하여, $s = F_a(i) \oplus h(x)$ 임을 확인한다.

무한 랜덤 스트링 P 대신에 의사 랜덤 함수 F를 사용하여 설계된 메시지 인증 방식의 안전성 역시 사용된 유니버설 해쉬 함수의 충돌 확률에 의존한다.

정리 5. *$H = \{h : A \rightarrow \{0,1\}^b\}$를 ε-AXU 해쉬 함수 패밀리라 하자. T_H는 랜덤 원소 $h \in H$를 계산하기 위하여 요구되는 시간이다. $T_h(q,u)$는 해쉬 함수 h를 이용하여 전체 길이가 μ 비트인 q개의 스트링의 해쉬값을 계산하는데 필요한 시간이다. $F : \{0,1\}^K \times \{0,1\}^l \rightarrow \{0,1\}^b$ 는 안전성이 $\varepsilon'(t,q)$인 의사 랜덤 함수이다. E를 t 시간 내에 전체 길이가 μ 비트인 q개의 query를 계산할 수 있는 공격자이며, E가 Wegman-Carter 패러다임에 대한 위장 공격에 성공할 확률을 δ라 하면,*

$$\delta \leq \varepsilon + \varepsilon'(t + \Delta t, q+1)$$

이다. 여기서, $\Delta t = O(T_h(q,\mu) + T_H + ql + qb)$ 이다.

III. Bucket 해쉬 함수

1. 정의

Bucket 해쉬 함수는 Crypto'95에서 Phillip Rogaway가 제안한 유니버설 해쉬 함수이다[10]. $X = X_1 \ldots X_n$들을 n개 워드(word)들의 스트링(string)이라 하자.Bucket 해쉬 함수의 기본 개념은 X의 각 워드들을 N개의 bucket에 분산시키고, 각 bucket에 저장된 내용들을 비트별 논리합 (bitwise exclusive-or) 하여, 그 결과들을 연접하여 해쉬값으로 설정하는 것이다.

B를 Bucket 해쉬 함수 패밀리라 하자. $w \geq 1$를 워드 크기, $n \geq 1$, $N \geq 3$으로 하고, $D = \{0,1\}^{wn}$을 정의역으로 $R = \{0,1\}^{wN}$을 치역으로 하는 해쉬 함수를 고려한다. 그러면, Bucket 해쉬 함수 패밀리를 3가지 파라미터 w,n,N에 의해 $B(w,n,N)$으로 표시할 수 있다. 예로서, $w=32$, $n=1024$, $N=140$을 들 수 있으며, 이것은 $B(32,1024,140)$이 된다.

Bucket 해쉬 함수 패밀리의 원소 $h \in B$는 집합 $\{1,2,\ldots,N\}$의 원소의 개수가 3인 서로 다른 부분집합 n개의 수열이다. 즉, $h \in B$에 대해서, $h = h_1 h_2 \cdots h_n (h_i \neq h_j)$이며, 각각의 h_i의 3개 원소는 $h_i = \{h_{i1} h_{i2} h_{i3}\}$로 표시할 수 있다.

Bucket 해쉬 함수 $h \in B$ 가 $X = X_1 \cdots X_n$을 해쉬 하는 과정은 다음과 같다.

- 단계 1 : $j = 1,2,\ldots N$에 대하여, $Y_j = 0$으로 설정한다.
- 단계 2 : $i = 1,2,\ldots n$에 대하여,
  - 단계 2-1 : $Y_{h_{i1}} = Y_{h_{i1}} \oplus X_i$
  - 단계 2-2 : $Y_{h_{i2}} = Y_{h_{i2}} \oplus X_i$
  - 단계 2-3 : $Y_{h_{i3}} = Y_{h_{i3}} \oplus X_i$
- 단계 3 : $Y_1 || Y_2 || \cdots || Y_N$을 해쉬값으로 출력한다.

Bucket 해쉬 함수에 의한 해쉬 과정은 다음으로 설명할 수 있다. 우리는 N개의 바구니를 가지고 있으며, 각각의 바구니는 초기에 비어 있다(단계 1). X의 첫 번째 워드 X_1을 h_1이 결정하는 3개의 바구니에 넣는다. 여기서, Bucket 해쉬 함수 $h=h_1h_2 \cdots h_n$의 $h_i(i=1,2, \ldots, n)$은 집합 $\{1,2, \ldots, N\}$의 원소의 개수가 3개인 부분 집합임을 상기하자. 즉, $h_1=\{3,6,9\}$라 하면, N개의 바구니 중 3번째, 6번째, 9번째 바구니에 첫 번째 워드 X_1을 넣는 것이다. 이러한 과정을 h_n 까지 계속해서 수행한다. 단계 2가 종료되면, N 개의 바구니에는 모두 $3n$개의 워드가 채워져 있다. 다음으로 각 바구니에 들어 있는 워드들을 각 바구니 별로 비트별 논리합을 취한다(단계 2). 각 바구니별로 비트별 논리합을 수행한 결과를 연접한 것을 해쉬값으로 출력한다(단계 3).

2. 충돌 확률(Collision probability)

Bucket 해쉬 함수 $h \in B[w,n,N]$의 충돌 확률(collision probability)은 모든 서로 다른 x, $x' \in \{0,1\}^{wn}$에 대해서, $h(x)=h(x')$이 될 확률의 최대값을 의미한다.

정리 6. *$w \geq 1$, $N \geq 32$, $\mathbf{n} \geq \binom{N}{3}/12$ 라고 가정하자. 그리고 ε을 $B[w,n,N]$의 충돌 확률이라 하자. 그러면, $\varepsilon \leq B(N)$ 이다. 여기서, $B(N)=\lambda(N)\,\beta(N)$이며,*

$$\lambda(N)=\frac{1}{1-\frac{6}{\binom{N}{3}}}$$

이고,

$$\beta(N)=\frac{720(N-3)(N-4)(N-5)+1944(N-3)(N-4)^2+648(N-2)(N-3)^2}{N^3(N-1)^3(N-2)^3}$$

이다.

정리 6에 의하면, $B[w,n,N]$은 $\varepsilon \approx 3312 \cdot N^6$인 ε-AU 함수이며, $N=140$인 경우의 충돌 확률은 $\varepsilon \approx 2^{-31}$이 된다. 일반적으로 유니버설 해쉬 함수의 충돌 확률은 2^{-30} 보다 작아야 하는 것으로 권고되고 있으므로, Bucket 해쉬 함수의 경우 이를 만족하기 위하여 $N \geq 140$이어야 하며, 워드 비트 길이가 32일 때 해쉬값의 비트 길이가 $w \cdot N=32 \cdot 140=4,480$ 비트로 매우 길기 때문에 추가적인 해쉬 과정이 요구되는 단점이 있다.

정리 6에서 충돌 확률 ε에 대해서, ε은 워드 비트 길이 w에 의존하지 않으며, $4 \leq n \leq \binom{N}{3}/12$인 경우에는, ε은 n에 의존하지 않는다.

$NN=N(N-1)(N-2)$이라 하면,

$$\lambda(N)=\frac{NN}{NN-36}$$

이다. $N \geq 32$라는 정리 6의 가정에 의해서, $1 \leq \lambda(N) \leq 1.002$이며, $\lambda(N)$을 무시할 수 있으므로, $B(N) \approx \beta(N)$이 된다.

정리 6의 조건들 중 $n \leq \binom{N}{3}/12$의 조건을 $n \leq \binom{N}{3}$으로 완화할 수 있으나, 현실적인 이점은 없다.

IV. LFSR 기반 유니버설 해쉬 함수

Crypto'94에서 Hugo Krawczyk는 LFSR (Linear Feedback Shift Register)을 기반으로 하는 유니버설 해쉬 함수를 제안하였다 [9]. Hugo Krawczyk이 제안한 유니버설 해쉬 함수는 Carter-Wegman의 랜덤 이진 행렬을 사용한 유니버설 해쉬 함수[11]의 개선이다.

먼저, Carter-Wegman의 랜덤 이진 행렬 기반 유니버설 해쉬 함수를 소개한다. A를 $n \times m$ 이진 행렬이라 하고 M을 길이가 m 비트인 메시지라 하자. $h_A(M)$을 행렬 A와 M 의 비트들로 구성된 열 벡터와의 이진 곱으로 정의한다.

$$h_A(M)=A \cdot M^T = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdots & \cdot \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix} \cdot \begin{bmatrix} M_0 \\ M_1 \\ \cdot \\ \cdot \\ \cdot \\ M_{m-1} \end{bmatrix}$$

Carter와 Wegman은 해쉬 함수 패밀리 $\{ h_A : A$는 $n \times m$ 이진 행렬$\}$가 $\Pr_{h \in H}[h_A(M)=c] \leq \varepsilon$인 ε-AU임을 증명하였다[11].

Carter-Wegman의 랜덤 이진 행렬 기반 유니버설 해쉬 함수는 m 비트 길이의 메시지를 해쉬하

기 위하여 $n \cdot m$ 비트 길이의 정보가 필요하다. Carter-Wegman의 랜덤 이진 행렬 기반 유니버설 해쉬 함수와 동일한 성질을 가지며, 해쉬 과정에 필요한 정보의 양은 보다 적게 할 수 있는 방법이 랜덤 이진 행렬 A를 사용하는 대신에 Toeplitz 행렬을 사용하여 설계 가능하다.

Toeplitz 행렬이란 각 대각열(diagonal)이 고정된 행렬이다. 즉, $1 \le i,k \le n$, $1 \le i,l \le m$에 대해서, $k-i=l-j$라면, $a_{ij}=a_{kl}$이다. 따라서, 행렬의 첫 번째 열과 첫 번째 행이 결정되면 Toeplitz 행렬의 모든 원소가 결정된다. 다음은 Toeplitz 행렬의 예이다.

$$\begin{bmatrix} 1 & 1 & 0 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 1 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 1 & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & 1 \\ \cdot & \cdot & \cdot & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

이러한 Toeplitz 행렬을 사용한 Carter-Wegman의 유니버설 해쉬 함수를 고려하자. 그러면, m 비트 메시지를 해쉬하기 위하여 모두 $n+m+1$ 비트의 정보가 필요하며, 이 양은 Carter-Wegman 유니버설 해쉬 함수에 비하여 훨씬 작은 양이다.

Hugo Krawzyck은 Toeplitz 행렬 기반 유니버설 해쉬 함수를 LFSR을 사용하여 개선하였다. 다음의 4×4 Toeplitz 행렬을 고려하자.

$$\begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

Toeplitz 행렬의 각 열들은 이전 열을 아래 방향으로 한 칸씩 쉬프트 시키고, 최상위 원소만을 새로운 값으로 설정한 것임을 쉽게 알 수 있다. 이러한 사실 때문에 Toeplitz 행렬을 LFSR로 대치한 유니버설 해쉬 함수를 고려할 수 있다. Hugo Krawzyck이 제안한 LFSR 기반 유니버설 해쉬 함수는 다음으로 구성된다.

- $p(x)$: 차수가 n인 $GF(2)$상의 기약 다항식 (irreducible polynomial)
- $s_0, s_1, \ldots$: 초기 벡터를 $\boldsymbol{s}=s_0,s_1,\ldots,s_{n-1}$로 하여 기약 다항식 $p(x)$를 특성 다항식 (characteristic polynomial)으로 하는 LFSR로 생성된 이진 수열

그러면, 임의의 n차 기약 다항식 $p(x)$와 초기 벡터 $\boldsymbol{s}\neq0$에 대해서, 유니버설 해쉬 함수 $h_{p,s}$를 다음으로 정의할 수 있다. 메시지 $M=M_0M_1\cdots M_{m-1}$에 대해서

$$h_{p,s}(M)=\bigoplus_{j=0}^{m-1} M_j \cdot (s_j, s_{j+1}, \cdots, s_{j+n-1}).$$

이러한 LFSR 기반 유니버설 해쉬 함수에서 m 비트 길이의 메시지 M를 해쉬하기 위하여 필요한 정보는 n차 기약 다항식과 n 비트 길이의 초기벡터이므로 모두 $2n$ 비트만이 필요하다.

정리 7. *$p(x)$를 $GF(2)$상의 차수가 n인 기약 다항식이라 하고, $\boldsymbol{s}=(s_0,s_1,\ldots,s_{n-1})$을 초기 벡터라 하자. 그리고, M을 길이가 m 비트인 메시지라 하자. $\lambda_1, \lambda_2, \ldots \lambda_n$을 $p(x)$의 서로 다른 근이라 하면,*

$$h_{p,s}=BD_{M,p}B^{-1}s^T$$

가 된다. 여기서, B는 $p(x)$를 기반으로하는 $n\times n$ 정칙 행렬(non-singular matrix)이며, $D_{M,p}$는 i 번째 대각열을 $M(\lambda_i)$로 가지는 $n\times n$ 대각 행렬이다.

위 정리를 이용하여, 참고 문헌[9]에서는 LFSR 기반 해쉬 함수가 ε-AU가 됨을 증명하였다.

정리 8. *LFSR 기반 해쉬 함수에 대해서,*

$$\Pr_{h\in H}[h_{p,s}(M)=c]\le\frac{m}{2^n-1}$$

이 된다.

V. MMH 유니버설 해쉬 함수

MMH^*_{32}와 MMH_{32}는 참고 문헌[11]에서 제안된 또 하나의 유니버설 해쉬 함수인 MMH^* (Multilinear-Modular-Hashing) 해쉬 함수의 개선으로 Shai Halevi와 Hugo Krawczyk이 제안한 것이다[8].

Carter와 Wegman은 참고 문헌[11]에서 어떤 소수 p에 대한 유한체 Z_p에서 동작하는 유니버설 해쉬 함수 MMH^*를 제안하였다. MMH^*는 어떤 정수 k에 대한 Z_p^k의 모든 다중 선형 함수 (multilinear function)들의 패밀리이다. p를 소

수, $k>0$을 임의의 정수라 하자. 그러면, MMH^*는 다음과 같이 Z_p^k를 정의역으로 하며, Z_p를 치역으로 하는 다음의 함수로 정의된다.

$$MMH=\{g_x : Z_p^k \to Z_p \mid x \in Z_p^k\}$$

g_x는 $\boldsymbol{x}=(x_1, x_2, \ldots, x_k)$와 $\boldsymbol{m}=(m_1, m_2, \ldots, m_k)$에 대해서,

$$g_x(\boldsymbol{m}) = \boldsymbol{m} \cdot \boldsymbol{x} \bmod p = \sum_{i=1}^{k} m_i x_i \bmod p$$

이다.

정리 9. *MMH^*에서, 모든 a와 모든 $\boldsymbol{m} \neq \boldsymbol{m}'$에 대해서,*

$$\Pr_{g_x \in MMH^*}[g_x(\boldsymbol{m}) - g_x(\boldsymbol{m}') = a \pmod p] = \frac{1}{p}$$

이다.

(증명). $a \in Z_p$라 하고, $\boldsymbol{m}$과 $\boldsymbol{m}'$을 $m_1 \neq m_1'$인 두 메시지라 하자. 그러면, 임의의 $x_2, \ldots, x_k$에 대해서 다음이 성립한다.

$$\begin{aligned}
&\Pr_{x_1}[g_x(m) - g_x(m') \equiv a \pmod p] \\
&= \Pr_{x_1}\left[(m_1 - m_1') \cdot x_1 + \sum_{i=2}^{k} m_i x_i \equiv a \pmod p\right] \\
&= \frac{\#\left\{x_1 \in Z_p \mid (m_1 - m_1') \cdot x_1 + \sum_{i=2}^{k} m_i x_i \equiv a \pmod p\right\}}{x_1\text{의 경우의 수}} \\
&= \frac{1}{p}.
\end{aligned}$$

□

MMH_{32}^*는 MMH^*의 변형으로 주요 개선 관점은 다음과 같다.

- 32 비트 정수의 처리를 기반으로한다.
- p를 $2^{32}+15$로 하여, division-less modular reduction을 구현할 수 있게 한다.

이에 따라, MMH_{32}^*는 다음으로 정의된다.

$$MMH_{32}^* = \{g_x : (Z_2^{32})^k \to Z_p \mid x \in (Z_2^{32})^k\}$$

g_x는 $\boldsymbol{x}=(x_1, x_2, \ldots, x_k)$와 $\boldsymbol{m}=(m_1, m_2, \ldots, m_k)$에 대해서,

$$\begin{aligned}
g_x(\boldsymbol{m}) &= \boldsymbol{m} \cdot \boldsymbol{x} \bmod (2^{32}+15) \\
&= \sum_{i=1}^{k} m_i x_i \bmod (2^{32}+15)
\end{aligned}$$

이다.

정리 10. *MMH_{32}^*에서, 든 a와 모든 $\boldsymbol{m} \neq \boldsymbol{m}'$에 대해서,*

$$\Pr_{g_x \in MMH^*}[g_x(\boldsymbol{m}) - g_x(\boldsymbol{m}') = a \pmod p] \leq \frac{1}{2^{32}}$$

이다.

(증명). $a \in Z_p$, $p=2^{32}+15$라하고, $\boldsymbol{m}$과 $\boldsymbol{m}'$을 $m_1 \neq m_1'$인 두 메시지라 하자. 그러면, 임의의 $x_2, \ldots, x_k$에 대해서 다음이 성립한다.

$$\begin{aligned}
&\Pr_{x_1}[g_x(\boldsymbol{m}) - g_x(\boldsymbol{m}') \equiv a \pmod p] \\
&= \Pr_{x_1}\left[(m_1 - m_1') \cdot x_1 + \sum_{i=2}^{k} m_i x_i \equiv a \pmod p\right] \\
&= \frac{\#\left\{x_1 \in Z_p \mid (m_1 - m_1') \cdot x_1 + \sum_{i=2}^{k} m_i x_i \equiv a \pmod p\right\}}{x_1\text{의 경우의 수}} \\
&= \frac{1}{2^{32}}
\end{aligned}$$

□

MMH_{32}는 MMH_{32}^*의 효율성을 개선하기 위한 목적으로 다음 사항을 고려하여 설계되었다.

- MMH_{32}^*의 출력을 32 비트로 한다.
- 내적 계산에서 64 비트 상위로 발생하는 캐리(carry)를 무시한다.

이에 따라 MMH_{32}는 다음으로 정의된다.

정의 4. *$p=2^{32}+15$이고, $k=32$라 하자. 그러면, MMH_{32}를 다음으로 정의한다.*

$$MMH_{32} = \{h_x : (Z_2^{32})^k \to Z_2^{32} \mid x \in (Z_2^{32})^k\}$$

여기서, 함수 h_x는 임의의 $\boldsymbol{x}=(x_1, x_2, \ldots, x_k)$와 $\boldsymbol{m}=(m_1, m_2, \ldots, m_k)$에 대해서, 다음으로 정의된다.

$$h_x(\boldsymbol{m}) = \left[\left[\left[\sum_{i=1}^{k} m_i x_i\right] \bmod 2^{64}\right] \bmod (2^{32}+15)\right] \bmod 2^{32}$$

MMH_{32}가 ε-AU 해쉬 함수가 됨을 보이기 위하여 먼저 다음 보조 정리를 증명한다.

보조정리 1. *$p=2^{32}+15$와, $k=32$에 대하여, H_{32}를 다음으로 정의하자.*

$$H_{32} = \{\widetilde{h}_x : (Z_2^{32})^k \to Z_p \mid x \in (Z_2^{32})^k\}$$

여기서, $\widetilde{h}_x$는 임의의 $x=(x_1, x_2, \ldots, x_k)$와 $m=(m_1, m_2, \ldots, m_k)$에 대해서, 다음으로 정의된

다.

$$\widetilde{h}_x=\left[\left[\sum_{i=1}^{k} m_i x_i\right]\bmod 2^{64}\right]\bmod(2^{32}+15)$$

그러면,

$$\Pr_{h_x\in H_{32}}[\widetilde{h}_x(m)-\widetilde{h}_x(m')\equiv a(\bmod p)]\le 2\cdot 2^{-32}$$

이 된다.

(증명) $a\in Z_p$ 그리고, $\boldsymbol{m}$과 $\boldsymbol{m}'$을 $m_1\neq m_1'$인 두 메시지라 하자. $x_1m_1<2^{64}$ 이므로, $x_2,\ldots,x_k$의 임의의 선택에 대해 x_1m_1은 $\left(\sum_{i=2}^{k}x_im_i\right)\bmod 2^{64}$에 많아야 하나의 캐리 비트를 추가한다. 따라서, $b(x_1)\in\{0,1\}$과 $b'(x_1)\in\{0,1\}$에 대해서, 다음으로 정리할 수 있다.

%

$$\left[\sum_{i=1}^{k}x_im_i\right]\bmod 2^{64}=s+x_1m_1-2^{64}\cdot b(x_1)$$
$$\left[\sum_{i=1}^{k}x_im'_i\right]\bmod 2^{64}=s'+x_1m'_1-2^{64}\cdot b'(x_1)$$

따라서,

$$\begin{aligned}&\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\\&=\left(\sum_{i=1}^{k}x_im_i\bmod 2^{64}\right)-\left(\sum_{i=1}^{k}x_im'_i\bmod 2^{64}\right)\\&=(m_1-m'_1)\cdot x_1-2^{64}\cdot(b(x_1)-b'(x_1))+s-s'\end{aligned}$$

이 된다. $b(x_1)$과 $b'(x_1)$의 값은 0 또는 1이다. 이를 $b(x_1)-b'(x_1)=-1$, $b(x_1)-b'(x_1)=0$, 그리고, $b(x_1)-b'(x_1)=1$의 3 경우로 나누면,

$$\begin{aligned}&\Pr_{x_1}[\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\equiv a(\bmod p)]\\&=\Pr_{x_1}[(m_1-m'_1)\cdot x_1-2^{64}\cdot(b(x_1)-b'(x_1))\\&\quad+s-s'\equiv a(\bmod p)]\\&\le\Pr_{x_1}[(m_1-m'_1)\cdot x_1\equiv a-s+s'-2^{64}(\bmod p)]\\&\quad(b(x_1)-b'(x_1)=-1\text{인 경우})\\&\quad+\Pr_{x_1}[(m_1-m'_1)\cdot x_1\equiv a-s+s'(\bmod p)]\\&\quad(b(x_1)-b'(x_1)=0\text{인 경우})\\&\quad+\Pr_{x_1}[(m_1-m'_1)\cdot x_1\equiv a-s+s'+2^{64}(\bmod p)]\\&\quad(b(x_1)-b'(x_1)=1\text{인 경우})\\&\le 3\cdot 2^{-32}\end{aligned}\qquad(1)$$

그러나, $b(x_1)-b'(x_1)$은 동시에 1 또는 -1의 값을 가질 수 없다. 왜냐하면, $b(x_1)=1$, $b'(x_1)=0$일 때, $x_1m_1+s\ge 2^{64}$이고, $x_1m'_1+s'<2^{64}$이므로, x_1m_1+s와 $x_1m'_1+s'$은 단조 증가 관계에 있게 된다. 따라서, $b(x_1)-b'(x_1)$은 동시에 1 또는 -1의 값을 가질 수 없게 된다. 그러므로, 식 (1)의 확률은 $2\cdot 2^{-32}$으로 된다. □

MMH_{32}는 H_{32}의 결과를 모듈러 2^{32}로 계산한 결과이다. 이를 이용하여 다음을 증명한다.

정리 11. MMH_{32}에서, 모든 a와 모든 $\boldsymbol{m}\neq\boldsymbol{m}'$에 대해서,

$$\Pr_{h_x\in MMH_{32}}[h_x(\boldsymbol{m})-h_x(\boldsymbol{m}')=a(\bmod p)]\le 6\cdot 2^{-32}$$

이다.

(증명). 고정된 임의의 값 $v(0\le v<2^{32})$와 두 개의 서로 다른 메시지 $\boldsymbol{m}$과 $\boldsymbol{m}'$, 그리고, $\boldsymbol{x}$를 $h_x(\boldsymbol{m})-h_x(\boldsymbol{m}')\equiv v(\bmod 2^{32})$가 되는 벡터라 하자. 그러면, $\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\equiv v(\bmod 2^{32})$ 이 된다. $\widetilde{h}_x(\boldsymbol{m})$과 $\widetilde{h}_x(\boldsymbol{m}')$는 모두 0과 $p-1$ 사이에 있으므로, $-p+1\le\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\le p-1$ 이 되며, $p=2^{32}+t$ 이므로,

$$-2^{32}-t+1\le\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\le 2^{32}+t-1$$

가 된다.

따라서, 다음을 얻는다.

$$\begin{aligned}&\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\\&=\begin{cases}\{v,v-2^{32}\}, & t\le v\le 2^{32}-t\\ \{v,v-2^{32},v+2^{32}\}, & 0\le v\le t-1\\ \{v-2^{32},v+2^{32}\}, & 2^{32}-t\le v\le 2^{32}-1\end{cases}\end{aligned}$$

즉, $\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')\equiv v(\bmod 2^{32})$ 이면, $\widetilde{h}_x(\boldsymbol{m})-\widetilde{h}_x(\boldsymbol{m}')$은 많아야 3가지 값만을 가진다. 따라서,

$$\Pr_{h_x\in MMH_{32}}[h_x(\boldsymbol{m})-h_x(\boldsymbol{m}')=a(\bmod p)]\le 3\cdot 2\cdot 2^{-32}$$

이 된다.

VI. 결론

참고 문헌[10]에서는 본 논문에서 소개한 유니버설 해쉬 함수들과 대표적인 해쉬 함수인 MD5의 효율성을 한 워드를 해쉬 하는데 필요한 명령이 수를 이용하여 비교하였다. 비교 내용은 표 1과 같다.

표 1. 제안된 여러 유니버설 해쉬 함수 및 해쉬 함수의 효율성 비교

종류	한 워드를 해쉬 하는데 필요한 instruction의 수
Bucket Hashing	10
LFSR-based hashing	24
MD5	35
MMH hashing	7.4

지금까지 본 논문에서는 유니버설 해쉬 함수의 정의 및 몇 가지 특성, 그리고, 유니버설 해쉬 함수를 이용한 인증 방식 설계 방법인 Wegman- Carter 패러다임을 소개하고, 지금까지 제안된 주요 유니버설 해쉬 함수의 내용을 기술하였다.

참고 문헌

[1] Mihir Bellare and Phillip Rogaway "Optimal asymmetric encryption". *Eurocrypt'94*, pp. 92-111. 1994.

[2] National Bureau of Standards. "FIPS PUB 46 : Data Encryption Standard, January 1977.

[3] Ronald L. Rivest. "The MD5 message digest algorithm". In *Request for Comments(RFC) 1321*, April. Internet Activities Board, Internet Privacy Task Force, 1992.

[4] Mihir Bellare, Ran Canetti, and Hugo Krawczyk. "Keying hash functions for message authentication". *Crypto'96*, pp. 1--15. 1996.

[5] Burt Kaliski and Matt Robshaw. "Message Authentication with MD5". *CryptoBytes*, volume 1, number 1, pages 5--8. 1995.

[6] Bart Preneel and Paul~C. van Oorschot. "MDx-MAC and building fast MACs from hash functions". *Crypto'95*, pp. 1--14. 1995.

[7] Mark N. Wegman and J.~Lawrence Carter. "New hash functions and their use in authentication". *Journal of Computer and System Sciences*, volume 22, pages 265--279, 1981.

[8] Shai Halevi and Hugo Krawczyk. "MMH: Software Message Authentication in the Gbit/second Rates". *Fast Software Encryption Workshop*, pp. 158--173, 1997.

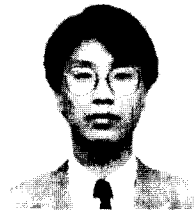
[9] Hugo Krawczyk. "LFSR-based Hashging and Authentication". In Yvo G. Desmedt, editor, *Crypto'94*, pp. 129--139. 1994.

[10] Phillip Rogaway. "Bucket hashing and its application to fast message authentication". *Crypto'95*, pp. 29--42. 1995.

[11] J. Lawrence Carter and Mark~N. Wegman. "Universal classes of hash functions". *Journal of Computer and System Sciences*, volume 18, pages 143--154, 1979.

著者紹介

박 상 우 (Sangwoo Park)



1989년 2월 : 고려대학교 수학교육과 졸업
1991년 8월 : 고려대학교 수학과 석사
1991년 8월 ~ 현재 : 한국전자통신연구원 선임연구원

지 성 택(Seongtaek Chee)

1985년 2월 : 서강대학교 수학과 졸업
1987년 8월 : 서강대학교 수학과 석사
1999년 2월 : 고려대학교 수학과 박사
1991년 8월 ~ 현재 : 한국전자통신연구원 선임연구원

박 춘 식(Choonsik Park)

광운대학교 전자통신과
졸업(학사)
한양대학교 대학원 전자통신과
졸업(석사)
일본 동경공업대학
전기전자공학과 졸업(암호학
전공, 공학박사)
1989년 10월 ~ 1990년 9월 일본 동경공업대학
객원 연구원
1989년 ~ 현재 한국전자통신연구원 책임연구원