通信情報保護學會誌
第9券 第3號 1999. 09

전산 정수론의 방법들을 이용하는 암호화 프로토콜에 대한 연구

김 철*

요 약

전산 정수론의 몇 몇 이론들과 그 계산 방법들은 최근 컴퓨터 및 네트웍의 발달로 그 유용성을 한층 증대 시키고 있다. 본 연구는 이들 전산 정수론의 방법들중에서 정수들의 집합 위에서의 다항식 계산의 복잡도를 이용하여 암호화 프로토콜에 응용하는 연구를 수행하여 그것을 암호학의 제 분야에서 활발히 활용되는 해쉬 알고리즘을 구성하였다. 또한 이러한 해쉬 알고리즘을 이용하여 암호화 프로토콜의 하나인 인증 프로토콜을 연구하였다.

먼저 group $SL_2(F_{2^n})$에 기초한 해쉬 함수에 대해 알아 본다. 행렬의 각 entry는 characteristic이 2인 finite field의 원소로 이루어져 있기 때문에 연산 속도도 매우 빠르며, 메시지의 작은 변조도 쉽게 감지 할 수 있다는 장점이 있다. 하지만 $SL_2(F_{2^n})$의 generator를 finite field의 원소로 표현하고, finite field F_{2^n} 상에서 discrete logarithm을 계산하면 충돌하는 메시지쌍이 존재하는 것을 알 수는 있으나, 이는 계산적으로 계산 불가능함을 증명하였다.

Ⅰ. 개 요

정보 교환이 빈번히 일어나는 정보 사회에서 정보의 인증은 아주 중요한 문제이다. 그리고 필요한 인증에는 메시지 인증, 사용자 인증 그리고 이 두 가지 기능을 합친 디지털 서명이 있다. 인증 방식에는 암호 기법을 효과적으로 이용하고 이를 구현하는 방법에는 일체형과 분리형 두 가지가 있다. 일체형은 메시지 전체를 암호화 처리를 하는 것으로 많은 계산을 요구하여 비효율적이라 할 수 있다. 반면, 분리형은 데이터를 암호화하기 이전에 압축하게 되는데 이때 쓰이는 압축 함수를 해쉬 함수라 한다. 즉, 해쉬 함수는 임의의 크기의 입력 비트 열에 대해 일정한 길이의 출력 비트 열을 내는 것으로 정보 보호의 여러 메커니즘에서 활발히 이용되는 요소 기술이다. 따라서, 인증 문제에 있어 메시지 압축이 주 기능인 해쉬 함수를 효율적으로 구성하는 것은 아주 중요하다. 해쉬 함수가 갖추어야 할 성질 중에서 가장 중요한 성질이 1-1 함수의 성질인데, 이것은 불가능하다. 그래서, [Jueneman, 1987]에는 해수 함수가 갖추어야 할 성질을 다음과 같이 설명되어 있다.

(1) 해쉬 알고리듬은 특별한 hardware의 구성없이 일반 computer 상에서도 효율적으로 실행되어야 한다.

(2) 메시지를 치환, 삭제, 삽입 등 조그마한 변조를

* 광운대학교 수학과, 신기술 연구소, kimch@daisy.kwangwoon.ac.kr, 이 논문은 96 대학 교수 해외 파견 연구 지원에 의하여 수행되었습니다.

가는 경우라도 해쉬 코드에서는 많은 변조가 있어야 한다.

(3) 서로 다른 두 개의 메시지를 압축했을 때, 두 개의 해쉬 코드가 같을 확률은 uniformly distributed random variable 이어야 한다.

(4) 해쉬 코드는 birthday attack을 피할 수 있을 정도로 길어야 한다.

(5) 해쉬 알고리듬은 invertible 이어서는 안된다.

지금까지 제안된 알고리듬은 안전도에 따라 두 개의 category로 나눌 수 있다. 즉, '알고리듬이 알려지더라도 같은 해쉬 코드를 서로 다른 두 개의 메시지 찾기가 계산상 불가능' 한 strong one-way frunction 과, '알고리듬이 알려져 있고 random하게 메시지가 주어지면 같은 해쉬 코드를 갖는 또 다른 메시지 찾기가 계산상 불가능' 한 weak one-way function이 그것이다. 여기서, 계산상 실행 가능성의 여부는 사용자의 특정한 보안 요구와 환경에 영향을 받는다. 그리고, 일반적으로 안전한 해쉬 함수라 함은 'strong' sense를 의미한다.

본 논문에서는 [4]와 [5]에서 제안된 idea를 바탕으로, 다음과 같은 안정성을 얻을 수 있는 disign principle을 따라 설계한 해쉬 함수를 제안한다. 즉, "입력 메시지의 작은 변조는 항상 탐지 할 수 있다." 그 외에도 여러 장점들이 있다. 즉, software로 쉽게 구현 할 수 있으며, parallelisation과 precomputation에 용이하며, 안정성은 어려운 수학 문제와 동치임을 보일 수 있다. 그러나, Asiacrypt '94에서 이 해쉬 함수에 대한 "attack" 이 제안되었다. 하지만, 이 공격 방법은 $SL_2(F_{2^n})$ 을 생성하는 행렬 A 와 B 의 order가 아주 작은 경우에 한했다. 그런, 논문 [1]에서는 행렬 A 와 B의 선택에 있어 암묵적으로 그런 경우는 배제했음이 언급되었다. 그 이후 field의 모든 representation에서 통용 가능한 공격법이 제안되었으며, 그 공격 방법은 행렬 A 와 B 로 생성되는 ring을 finite field로 embeding 하는 것에 기초를 두었다. 이 방법으로 충돌 메시지 쌍을 찾는 것은 F_{2^n} 혹은 $F_{(2^n)^2}$ 상에서 discrete logarithm을 계산하는 것 만큼 어렵다. 그리고, 이 공격 방법으로 찾은 충돌 메시지 쌍의 형태를 보면, 0 또는 1로만 된 긴 메시지 형태이기 때문에, 실제로 공격법인 양 보이진 않는다.

Ⅱ. SL_2 해쉬 함수

2.1 Design Principle and Graph Theoretic Issues

SL_2 해쉬 함수는 다음의 일반적인 형태를 따른 것이다.

(1) Defining Parameter.

(a) A : 알파벳.

(b) G : order가 유한인 group.

(c) S : order가 A 의 order와 같고, group G 의 generator를 원소로 하는 집합.

(d) ϕ : $A \rightarrow S$ 1-1 함수.

(2) 알고리듬

메시지 $x_1 x_2 \cdots x_k$ 의 해쉬 코드는 group G 의 원소 $\phi(x_1)\,\phi(x_2)\cdots\phi(x_3)$ 로 한다.

위와 같이 일반적인 형태를 따른 이유는 다음과 같은 두 가지 성질 때문이다.

(1) 연접 성질 : 만약 x 와 y 를 두 개의 메시지라면, 두 메시지의 연접 $x \| y$ 의 해쉬 코드는 $\phi(x \| y) = \phi(x)\,\phi(y)$ 이다. 이 성질은 parallelisation과 precomputation에 용이하다.

(2) 안전성 문제를 다음의 Cayley graph와 관련시켜 다룰 수 있다.

Parameter of the associated Cayley graph $C(G, S)$: Cayley graph $C(G, S)$ 의 vertex 집합은 G 이고, 두 개의 vertex v_1 과 v_2 에 대해 $(v_1)^{-1} v_2 \in S$ 이면 v_1 에서 v_2 로 방향이 주어진 edge가 존재하는 graph이다. 여기서, G와 S 는 defining parameter에 나오는 group과 generator 집합이다.

정의 2.1 graph G 가 주어지면 directed girth ρ 를 다음과 같이 정의한다. 임의로 주어진 vertex v_1 과 v_2 에 대해, v_1 에서 v_2 로의 directed path들이 주어진다면 path의 길이 중 가장 작은 것을 λ_{v_1, v_2} 라 하자.

그러면 $\rho = \min_{v_1, v_2 \in G} \lambda_{v_1, v_2}$ 로 정의한다.

위의 정의를 다음과 같은 해쉬 함수의 성질로 번역할 수 있다.

성질 2.2 만약 메시지 $x = x_1x_2\cdots x_ix_{i+1}\cdots x_{i+k}$ $x_{i+k+1}\cdots x_t$ 와 x 의 연속된 k 개의 symbol이 연속된 h 개의 다른 symbol로 대치된 $y = x_1x_2\cdots x_i$ $y_{i+1}\cdots y_{i+h}x_{i+k+1}\cdots x_t$ 가 같은 해쉬 코드를 갖게 된다면 $\max(k, h) \geq \rho$ 를 만족해야 한다.

즉, 아주 큰 girth ρ 를 갖는 Cayley graph $C(G, S)$ 를 얻을 수 있다면 위의 설질에 의해 메시지 일부분의 변조를 막을 수 있다. Cayley graph와 연관시켜 얻을 수 있는 성질 가운데 중요한 성질이 다음에 있다.

성질 2.3 Cayley graph $C(G, S)$ 에 있는 cycle의 length들의 최대공약수가 1이면, 해당하는 해쉬 함수에 대해, n 이 커질수록 길이 n 인 메시지의 해쉬 코드 분포는 균일 분포에 가까워진다.

2.2 The choice of SL_2

Finite field의 원소를 entry로 하는 SL_2 의 원소를 해쉬 코드로 정의한 데는 다음과 같은 연유에서이다. 우선, 간단한 행렬을 generator로 사용함으로써 해쉬 코드를 쉽게 얻을 수 있다. 그리고, 큰 값의 girth를 갖는 group상에서 정의된 Cayley graph를 비교적 쉽게 얻을 수 있다.[3][6]

2.3 A specific Hash function

이 section에서는 SL_2 해쉬 함수의 defining parameter와 알고리즘을 알아보고 SL_2 해쉬 함수가 가지고 있는 안전성과 관련된 다음의 세 가지에 대해 알아본다.

(1) 해쉬 코드의 집합은 무엇인가?

(2) 메시지 일부분의 변조는 어떻게 감지할 수 있는가?

(3) density attack을 어떻게 피할 수 있는가?

2.3.1 Definging parameter and 알고리즘

1. Defining Parameter.

(1) 정수 n 을 130과 170 사이에서 택한다.

(2) $F_2[x]$ 에서 degree가 n 인 irreducible polynomial $P_n(x)$ 를 택한다.

여기서, n 을 130 - 170 사이의 정수로 택한 것은 이 범위에서 paralleization과 precomputation이 가장 용이하기 때문이다.

2. 알고리즘

(1) $A = \begin{pmatrix} x & 1 \\ 1 & 0 \end{pmatrix}$ $B = \begin{pmatrix} x & x+1 \\ 1 & 1 \end{pmatrix}$

(2) 함수 $\varnothing : \{0, 1\} \rightarrow \{A, B\}$ 를 다음과 같이 정의한다.

$$\varnothing(0) = A, \quad \varnothing(1) = B$$

(3) binary message $x_1x_2\cdots x_k$ 의 해쉬 코드는 행렬 곱인 $\varnothing(x_1)\varnothing(x_2)\cdots\varnothing(x_k)$ 로 정의한다. 여기서, 각 entry의 계산은 quotient field $F_{2^n} = F_2[x]/P_n(x)$ 에서 한다.

2.3.2 해쉬 코드의 집합

여기서는 다음 정리를 이용하여 해쉬 코드의 집합이 전체 SL_2 가 됨을 보이겠다.

정리 2.4 $SL_2(F_{2^n})$ 의 가능한 진부분군(proper subgroup)의 형태는 다음과 같다.

(1) Abelian groups.

(2) Dihedral subgroups of order $2d$, d는 $2^n + 1$ 또는 $2^n - 1$ 의 약수.

(3) A_4, A_5 또는 symmetric group S_4.

(4) The upper triangular subgroup, its subgroups, and their conjugates.

(5) $SL_2(F_{2^m})$, 여기서 m 은 n 의 약수.

증명 위의 다섯 가지 어느 형태도 가능하지 않음을 보일 수 있다[1]. ∎

2.3.3 On the difficulty of finding collisions

해쉬 코드를 group의 원소로 정의한다면 collision을

찾는 문제는 다음과 같이 두 개의 동치인 수학 문제로 표현할 수 있다.

(1) $s_1 s_2 \cdots s_n = \sigma_1 \sigma_2 \cdots \sigma_m$ 을 만족하는 group G 의 gmerator $s_1, s_2, \ldots, s_n, \sigma_1, \sigma_2, \ldots, \sigma_m$ 찾기

(2) group G 의 identity 1의 nontrivial factorization 찾기

위의 두 번째 문제와 관련해서 언급할 수 있는 내용은, 일반적으로 모든 유한 group에서는 항상 trivial factorization이 존재한다. 즉, group G 가 유한 group이라면 S 의 모든 원소 s 에 대해 $s^{|G|} = 1$ 이 성립한다. 하지만, 모든 해쉬 코드를 표현하기 위해 필요한 비트는 $n \simeq \log G$ 이다. 따라서, group G 의 order를 아주 크게 잡는다면, 예를 들어 $|G| = 2^{500}$ 로 잡는다면, 일반적으로 2^{500}bit 크기의 메시지가 존재하지 않기 때문에, trivial factorization을 이용해 메시지를 변조한다는 것은 의미가 없다 하겠다. 더욱이, group G 의 identity 1의 nontrivial factorization이 계산상 불가능하다면 strong collision criterion을 만족하게 된다. group G 를 $SL_2(F_q)$ 로 택했을 때 identity 1의 short factorization을 찾기 위한 시도들이 많이 있었다. 그 결과들을 다음에 요약해 본다.

(1) 부분군 공격(subgroup attack)
이 방법은 hashcode가 부분군을 이루게 되는 메시지를 찾는 방법이다. 일반적으로 $SL_2(F_q)$ 는 index를 $q+1$ 보다 크게 가진다. 하지만, 우리는 q 를 아주 크게 잡은 상태이므로 이 방법은 비효율적이다.

(2) density attack
$SL_2(F_q)$ 상에서 정의된 해쉬 함수가 제안되기 이전에 $SL_2(Z_p)$ 에 기초한 해쉬 함수가 불안전함을 보였다[11]. 이 경우, $SL_2(Z_p)$ 는 두 개의 generator $C = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}$ 와 $D = \begin{pmatrix} 1 & 0 \\ 1 & 1 \end{pmatrix}$ 를 갖는다. 이 해쉬 함수에 대한 공격 또한 identity의 short factorization을 찾는 방법으로, 우선 identity를 무한 group에서, 이 경우에는 $SL_2(Z)$ 에서, 각 entry를 modulo p 연산을 했을 경우 항등 행렬이 되는 행렬 U 를 찾는 것이다. 물론 U 는 C 와 D 의 곱으로 표현되어야 한다. 이 경우, 행렬 U 는 음이 아닌 entry를 가져야 한다. 하지만, 음이 아닌 값을 entry로 하는 행렬들은 $SL_2(Z)$의 'dense'한 subset을 이룬다. 따라서, 이 공격 방법을 피하기 위해서는 $SL_2(F_q)$ 와 관계있는 무한 group에서 'sparse' 한 submonoid를 형성하는 group의 generator 집합 S 가 필요하다. 효과적인 변조를 위해서는 C 와 D 의 짧은 곱으로 된 행렬 U 가 필요한데, 이를 위한 probabilistic algorithm이 [7]에 나와 있다.

Ⅲ. Attack using involutions

SL_2 해쉬 함수에 대한 공격 방법으로 group 이론에 나오는 "involution" element를 이용할 수 있다. "involution" 원소란 order가 2인 원소를 지칭하고, 다음과 같이 $SL_2(F_q)$의 generator A 와 B 를 이용해 involuion 원소들을 다음과 같이 만들 수 있다.

보조정리 3.1 $\alpha = A^{-1}B$ 라 하면 α 는 involution이다.

보조정리 3.2 $\beta = A\alpha A^{-1}$ 라 하면 β 는 involution이고, $\alpha\beta \neq \beta\alpha$.

따라서, α와 β 로 생성된 $SL_2(F_q)$ 의 부분군은 Dihedral group과 isomorphic하다.
다음의 보조정리는 $SL_2(F_q)$ 에서 성립하는 것이고 A 와 B 의 order만 결정된다면 충돌하는 메시지쌍을 쉽게 구할 수 있겠다.

보조정리 3.3 두 개의 involution α와 β 는 다음을 만족한다.

$$\alpha(\alpha\beta)\alpha = \beta\alpha$$

정리 3.4 $SL_2(F_q)$ 에서는 다음의 관계식을 만족한다.

$$A^{-1}B(A^{-1}B^2A^{-1})A^{-1}B = BA^{-2}B.$$

Ⅳ. 행렬 표현을 이용한 attack

4.1 Finite Field의 행렬 표현

이 절에서는 finite field의 기초적인 성질을 알아본다. 더 자세한 것은 [8]과 [2]를 참조하기 바란다.

정의 4.1 F_{q^n} 의 한 원소 β 에 대해 F_q 상에서의 trace 값은 다음과 같이 정의한다.

$$tr(\beta) = \sum_{i=0}^{n-1} \beta^{q^i}$$

trace 함수는 일반적으로 dual basis를 정의할 때 이용한다.

정의 4.2 $A = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ 을 F_q 상에서의 F_{q^n} 의 basis라 하자. 그러면, A 의 dual basis $\{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ 은 다음을 만족하는 것이다.

$$tr(\alpha_i \beta_j) = \delta_{ij}.$$

여기서, i와 j는 0과 $n-1$ 사이의 정수이다.

finite field의 모든 basis는 unique한 dual basis를 가지며, dual basis의 개념은 finite field를 행렬 표현(matrix representation0하는데 중요한 tool로 사용된다.

정의 4.3 $A = \{\alpha_0, \alpha_1, \ldots, \alpha_{n-1}\}$ 을 F_q 상에서의 F_{q^n} 의 basis라 하고, $\{\beta_0, \beta_1, \ldots, \beta_{n-1}\}$ 을 dual basis라 하자. 그러면, A 에 관한 행렬 표현 I_A 는 다음과 같이 정의한다.

$$I_A : F_{q^n} \to F_q^{n \times n},$$

즉, $I_A(\gamma) = (tr(\gamma \alpha_i \beta_j))\ 0 \le i, j \le n-1$

4.2 Embeding

여기서 소개하는 공격 방법은, $SL_2(F_{2^n})$ 의 generator인 A 와 B 를 finite field에 embeding 시키는 것에 기초한다.

참고 4.4 행렬 A 는 다항식 $\tilde{A}(x) = x^2 + \alpha x + 1$ 의 근이 되고, 행렬 B 는 $\tilde{B}(x) = x^2 + (\alpha + 1)x + 1$ 의 근이 된다.

위의 참고에 의해 행렬 A 와 A 의 멱승들은 F_{2^n} 의 extension field $F_A (F_A \subset F_{2^n}^{2\times 2})$ 의 원소가 된다. F_A 의 수학적인 구조를 보기 위해서는 A 가 만족하는 방정식, 즉, $x^2 + \alpha x + 1 = 0$ 의 근을 F_A 에서 찾아야 하겠다. 다항식 $\tilde{A}$ 가 irreducible인 경우는, $F_{(2^n)^2}$ 의 행렬 표현을 이용하면 된다. 일반성을 잃지 않고, 다항식 $\tilde{B}$ 에 대한 결론도 $\tilde{A}$ 와 같은 결론이 나오므로 다항식 $\tilde{A}$ 에 관한 내용만 살펴본다.

4.2.1 The irreducible case

$\tilde{A} = x^2 + \alpha x + 1 \in F_{2^n}[x]$ 가 irreducible인 경우에 대해 살펴보자. 이 경우, $F_{(2^n)^2} \simeq F_{2^n}[x] / \tilde{A}$ 이고 basis $\Omega = \{x, 1\}$ 에 관한 행렬 표현 $I_\Omega(1) = I$ (항등 행렬)을 함수값으로 갖게 된다. 따라서, I_Ω 에 의해 $F_{2^{n\cdot 2}} \simeq F_{2^n} \cdot A + F_{2^n} \cdot I$ 는 2×2 행렬로 이루어진 ring으로 embeding 된다. $SL_2(F_{2^n})$ 의 원소들은 determinant가 항상 1이다. 그리고, $F_1 : SL_2(F(2^n)) \cap (F_{2^n} \cdot A + F_{2^n} \cdot I)$ 의 원소 개수는 $2^n + 1$ 개 이다.

따라서, $F_{2^n} \cdot A + F_{2^n} \cdot I$ 을 같은 determinant 값을 갖는 $2^n - 1$ 개의 집합으로 separate 할 수 있다. 더욱이, A 의 order는 $2^n + 1$ 의 약수이다. 이상의 내용으로 다음 정리를 얻을 수 있다.

정리 4.5 $SL_2(F_{2^n})$ 의 원소 $A = \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}$ 가 주어지고, $\tilde{A} : x^2 + \alpha x + 1$ 을 irreducible 이라 하자. 그러면, $F_{2^n}^{2\times 2}$ 의 원소인 M 이 A 의 멱승이기 위한 필요충분 조건은 다음 세 조건을 동시에 만족하는 것이다.

(1) $M = \lambda A + \mu I$, 여기서 $\lambda, \mu \in F_{2^n}$.

(2) $|M| = 1$

(3) $M^{ord(A)} = I$.

4.2.2 The reducible case

다항식 $x^2 + \alpha x + 1$ 이 reducible이라 하면, 즉 F_{2^n} 의 원소 β 가 존재하여 $(x + \beta)\left(x + \frac{1}{\beta}\right) = x^2 + \alpha x + 1$ 을 만족한다.

참고 4.6 $V := F_{2^n} \cdot A + F_{2^n} \cdot I$ 에 행렬 곱 연산을 정의하고, F_{2^n} 위에서 2차원 vector space로 봤을 경우, V 는 abelian F_{2^n} — algebra가 된다.

보조정리 4.7 $M_1 = \lambda_1 A + \mu_1 I$, $M_2 = \lambda_2 A + \mu_2 I$ 라 하자. 여기서 $\lambda_1, \lambda_2, \mu_1, \mu_2 \in F_{2^n}$. 그리고, F_{2^n} 의 한 원소 β 를 $x^2 + \alpha x + 1$ 의 한 근이라 하자. 그러면, 다음의 relation "~" 은 algebra V 에서 equivalence relation이 된다.

$\varnothing : V/\sim \rightarrow F_{2^n}$, $\varnothing\,\overline{(\lambda A + \mu I)} = \lambda\beta + \mu$.

여기서, $\overline{\lambda A + \mu I}$ 는 $\lambda A + \mu I$ 를 포함하는 equivalence relation이다. 그리고, $\overline{0}$ 이 아닌 $V/\sim$ 의 한 원소 $\overline{M_0}$ 와 F_{2^n} 의 한 원소 γ 에 대해, $\overline{M_0}$ 의 한 원소 M 이 유일하게 존재하여 $|M| = \gamma$ 가 된다.

증명 참고[9]. ∎

정리 4.8 $SL_2(F_{2^n})$ 의 원소 $A := \begin{pmatrix} \alpha & 1 \\ 1 & 0 \end{pmatrix}$ 에 대해 거듭 제곱으로 표현되기 위한 필요충분 조건은 다음의 세 조건을 동시에 만족하는 것이다.

(1) $M = \lambda A + \mu I$. 여기서, $\lambda, \mu \in F_{2^n}$

(2) $|M| = 1$.

(3) $(\lambda\beta + \mu)^{ord(A)} = 1$.

증명 참고[9]. ∎

4.3 충돌 메시지쌍 찾기

앞 절에서 살펴본 결과들을 이용해 SL_2 해쉬 함수의 충돌 메시지 쌍을 찾아보자, 알고리즘은 다음과 같다.

(1) $SL_2(F_{2^n})$ 의 임의의 원소 C 에 대해 다음과 같은 방정식을 세운다.

$$A^{i_1}B^{i_2}\cdots A^{i_l} = (\lambda_1 A + \mu_1 I)(\lambda_2 B + \mu_2 I)\cdots (\lambda_l A + \mu_l I) = C$$

여기서, l 은 3 또는 4 정도로 작게 잡는다.

(2) 위의 식으로부터 l 차 방정식 4개를 얻을 수 있다. 그리고, $|\lambda_i A + \mu I) = 1$ 임을 이용해 l 개의 방정식을 더 구할 수 있다. 실제로 $l = 3$ 인 경우, 대부분 이 연립 방정식의 근은 존재한다. 만약 $l = 3$ 인 경우 근이 존재하지 않으면, C 을 다른 것으로 택하거나 제 4의 행렬을 택하여 새로이 식을 구한다.

(3) A 와 B 의 order에 관한 조건들을 체크한다. 만약 조건을 만족시키지 못하면, 다시 1단계로 돌아간다. 조건을 만족시키면, 행렬에 관한 discrete logarithm을 계산한다.

이 공격 방법의 complexity 또한 A 와 B 의 order에 의존한다. order가 작은 경우, 마지막 step이 성공할 확률은 작다. 따라서, 방정식 A 와 B 에 관한 관계식이 더 추가되어야 한다. 어쨌든, discrete logarithm이 complexity를 좌우한다고 생각된다. 이 절에서 제시한 알고리즘을 따라 충돌하는 메시지 쌍을 찾을 수 있고, 몇 개는 underlying field에 관계없이 일반적으로 성립한다. 다음 식은 irreducible polynomial $p(x) = x^{21} + x^2 + 1$ 을 이용한 field F_{21} 에서 "AXIOM"을 이용해 구한 식이다.

$$A \cdot B^{-1} = B \cdot A^{-1}$$

$$A^2 \cdot B^{-3} = B^3 \cdot A^{-2}$$

$$A \cdot B \cdot A^{-4} \cdot B^2 = B^3 \cdot A$$

$$A \cdot B \cdot A^{-3} \cdot B^2 = B \cdot A^2$$

$$A^2 \cdot B \cdot A^{-2} \cdot B^4 = B^{-1} \cdot A^2$$

V. 결 론

이상으로 행렬 구조를 이용한 해쉬 함수에 대해 알아 보았다. 제안된 공격 방법들이 모두 A 와 B 의 order에 상당히 의존하고 있다. 따라서, order에 관한 A 와 B 의 확률 분포등에 관한 연구 등, 안전성과 관련된 몇 가지 사실만 입증한다면, 공격에 대해 약간의 취약성을 보이긴 했지만, 그 외에 증명이 된 안전성에 관한 여러 성질이 상당한 장점으로 꼽을 만 하다.

VI. Appendix

여기서는, $n = 130$ 인 경우 irreducible polynomial $P_{130}(x) = x^{130} + x^3 + 1$ 을 이용한 해쉬 암수의 결과를 알아 본다. 결과들은 16진수 표현임을 말해 둔다.

(1) Message = a

$$Hashcode = \begin{pmatrix} 3b & 57 \\ 29 & 6a \end{pmatrix}$$

(2) Message = A

$$Hashcode = \begin{pmatrix} 29 & 6a \\ 12 & 3d \end{pmatrix}$$

(3) Message = abcdefghijklmnopqrstuvwxyz

$$Hashcode = \begin{pmatrix} 3d0a3872769583 & 55b384f5e45382 \\ 2dadfc9383d204 & 695e3d6672c187 \end{pmatrix}$$

(4) Message = abcdefghijklmnopqrstuvwxyz

$$Hashcode = \begin{pmatrix} da9791b640a99 & 11ae9ee13eb7b8 \\ 856e6ff5ea820 & 1df8781a255ad9 \end{pmatrix}$$

(5) Message = ABCDEFGHIJKLMNOPQRSTUVWXYZ

$$Hashcode = \begin{pmatrix} f3e34a7611 & 18bc72234f1 \\ 6c213c52f1 & e7a20e2300 \end{pmatrix}$$

(6) Message = Abcdefghijklmnopqrstuvwxyz

$$Hashcode = \begin{pmatrix} 2dadfc9383d204 & 695e3d6672c187 \\ 10a7c4e1f54787 & 3cedb993969205 \end{pmatrix}$$

(7) Message = Abcdefghijklmnopqrstuvwxyz

$$Hashcode = \begin{pmatrix} 856e6ff5ea820 & 1df8781a255ad9 \\ 5ff9fe43ad2b9 & c56e6fb1bed61 \end{pmatrix}$$

(8) Message = bcdefghijklmnopqrstuvwxyzABC DEFGHIJKLMNOPQRSTUVWX YZ

$$Hashcode = \begin{pmatrix} b0fac922009781180821a1b7 & 123718d78c07de6dde97f93d \\ c28e151ec1fa66c5f1018db29 & 1f5eadc234093805c192a0f2 \end{pmatrix}$$

참 고 문 헌

[1] J-P. Tillich and G.Zemor. Hashing with SL_2 . Proceedings of Crypto '94, Yvo G.desmet(ED.) LNCS Vol.839, Sppringer -Verlag, pp.40-49, 1994.

[2] W.Geiselmann, D. Gollmann; Self-Dual basis in F_{q^n} ; Design, Codes and Crypto -graphy, Vol.3, No.4, pp.333-345,1993.

[3] J.Lafferty, Rockmore, D : Numerical investigation of the spectrum for certain families of Cayley graphs. In 1992 DIAMACS Workshop on expanders graphs(1993) D.D. in Disc. Math., T.C.S. in Discrete Mathemaics, and T. C. Science, Eds.pp.63-74.

[4] G.Zemor : Hash functions and Cayley graphs. To be appeared in Designs, Codes and Cryptography.

[5] G.Zemor : Hash functions and graphs with large girths. In Eurocrypt 91(1991) LNCS 547, Springer-Verlag, pp.508-511.

[6] Sarnack, P : Some applications of modular forms. Cambridge University Press 1990.

[7] J-P. Tillich, G.Zemor : Group-theoretic hash functions. In First French-Israeli Workshop on algebraic coding(1994) Springer-Verlag

Lec. N. Comp. Sci. 781 pp.90-110
[8] R.Lidl, H. Niedrreiter : Introduction to Finite Fields and their applications ; Cambridge University Press, 1986.
[9] Willi Geiselmann : A Note on the Hash Function of Tillich and Zemor.

□ 著者紹介

김철(Chul Kim) **정회원**

1984년 2월 : 연세대학교 수학과 학사
1984년 3월 : 연세 대학교 대학원 수학과 입학
1989년 12월 : North Carolina 주립대 대학원 수학과 석, 박사

<관심분야> 암호학, 부호이론, 모의실험 이론, 계산이론, 응용 대수학