이차원 셀룰라 오토마타에 기반하는 해쉬 함수

*김재겸

요약

본 연구에서는 새로운 구조의 이차원 셀룰라 오토마타를 제안하고 제안된 이차원 셀룰라 오토마타에 기초한 해쉬 함수를 제안하고 제안된 해쉬 함수의 통계적 검증 결과를 밝힌다.

A hash function based on 2D cellular automata

*Jae-gyum Kim

Abstract

In this paper, We proposed a hash function based on the concept of 2-dimensional cellular automata which is a method of analyzing dynamical systems. The proposed hash function is designed for using and disusing key. And the key size is variable from 128 bits.

1. 서론

본 연구의 목적은 안전하면서 그 수행속도가 빠른 전용 해쉬 함수를 개발하는 것으로 인터넷의 발전은 21세기 e-business 시대로의 새로운 전환을 가져왔고, 인터넷의 효율적인 활용이 모든 분야와 조직에서 경쟁력을 좌우하고 있어, 인터넷의 특성인 개방화 및 분산화 된 환경에서 각종 전자상거래, 전자문서, 전자우편 등에 이용되는 정보와 인터넷에 접속된 시스템과 LAN 등의 네트워크 보호를 위한 각종 보안기술에 대한 중요성이 날로 부각되고 있다.

이 논문은 2001학년도 경 성대학교 학술지원연구비에 의하여 연구되었음

* 경성대학교 수리과학부

이러한 보안기술의 핵심적인 요소 중의 하나가 안전하면서도 속도가 빠른 전용 해쉬 함수이다.

해쉬 함수란 임의의 크기의 메시지를 입력으로 받아 고정된 크기의 출력값을 생성해내는 함수로 해쉬값이 입력 스트링의 압축된 대표 이미지로 작용하고, 그 스트링을 유일하게 식별할 수 있는 것으로 사용될 수 있다는 점에 의하여 사용자 인증, 전자화폐, 전자서명, 공개키 암호 알고리즘 등의 여러 분야에서 사용하고 있다.[1]

해쉬 함수는 입력 파라미터의 종류에 따라 하나의 입력 파라미터-메시지-를 가지는 unkeyed 해수 알고리즘과 두 개의 다른 입력 파라미터-메시지와 비밀키-를 가지는 keyed 해쉬 함수의 두 분류로 나누어 질 수 있으며, 해쉬 함수의 전체 구성에 따라 DES와 같은 블록 암호 알고리즘에 기초한 해쉬 함수와 전용 해쉬 함수로 나눌 수 있다. 블록 암호 알고리즘을 이용한 해쉬 함수는 이미 구현되어 사용되고 있는 블록 암호 알고리즘을 사용할 수 있다는 이점이 있으나, 대부분의 블록 암호알고

리즘의 속도가 그리 빠르지 않을뿐더러 이를 기본 함수로 이용한 해쉬 함수의 경우 블록 암호보다도 훨씬 더 속도가 떨어지므로 현재는 대부분의 응용에서 전용 해쉬 함수가 주로 이용된다.

국내에도 HAS160 이라는 전용 해쉬 함수를 표준으로 사용하고 있다.

1985년 동역학계(Dynamical Systems)의 일종인 CA(Cellular Automata)를 이용한 암호 시스템이 Wolfram에 의해서 최초로 제안된 후[2], 1994년 Nandi 등에 의해서 PCA(Programmable Cellular Automata)를 이용한 암호 시스템들이 제안된 바 있고[3], 1997년 Mihaljević에 의해서 Nandi 등의 시스템을 개선한 암호 시스템이 제안된 바 있다[4].

CA는 그 본질적인 특성이 확산(Diffusion)과 국소적인 상호 작용(Local Interaction)이므로 암호 시스템과 VLSI 환경에 적합한 것으로 알려져 있다.

CA는 셀들의 배열의 1차원 구조에 기초한 1차원 CA와 셀들의 배열의 2차원 이상의 다차원 구조에 기초한 다차원 CA로 나눌 수 있는데, 다차원 CA는 그 구조의 복잡성으로 인해 분석이 어렵고, 각 셀의 상태가 다른 셀들의 상태에 영향을 주는 속도가 1차원 CA에 비하여 훨씬 더 빠르므로 1차원 CA보다 암호 시스템에 적합하다. 그러나 현재까지의 CA의 암호 시스템에의 응용에 관한 연구는 주로 1차원 CA에 머물러 왔는데, 이는 1차원 CA의 이론적인 연구의 용이성 때문이다.

본 연구에서는 새로운 구조의 이차원 CA를 제안하고 이를 이용한 전용 해쉬 함수를 제안한다.

본 논문의 구성은 다음과 같다. 2절에서는 해쉬 함수에 관한 정의와 일반적 모델에 관해, 3절에서는 셀룰러 오토마타에 관한 기본 이론을 기술한다. 4절에서는 새로운 구조의 이차원 CA를 제안하고 5절에서는 이 구조를 이용한 전용 해쉬 함수를 제안하고 제안된 전용 해쉬 함수의 통계적 특성에 기초한 안전성을 분석하며 마지막으로 6절에서는 결론을 맺는다.

2. 해쉬 함수의 정의와 일반적 모델

해쉬 함수(hash function) 즉, 암호학적 해쉬 함수(cryptographic hash function)는 임의의 유한 길이 비트 스트링을 고정된 길이의 스트링으로 사상시키는 함수이다. 이 출력은 흔히 해쉬값(hash value), 메시지 다이제스트(message digest) 등으로 불린다. 암호학적으로 안전한 해쉬 함수는 함수 h와 입력 x가 주어지면, 해쉬값 h(x)를 계산하는 것은 쉬워야 하지만 함수 h와 해쉬값 h(x)로부터는 입력 x를 찾는 것이 계산상 불가능해야 한다는 일방향성을 가져야 한다.

일방향 해쉬 함수는 다음 성질을 만족해야 한다.

- preimage resistance : 해쉬값 y가 주어졌을 때, h(x)=y를 만족하는 입력 x를 발견하는 것이 계산상 수행 불가능하다.

-2nd preimage resistance : 입력 x와 출력 h(x)가 주어졌을 때, h(x)=h(x′)을 만족하는 입력 $x \neq x'$를 발견하는 것이 계산상 수행 불가능하다.

암호학적으로 유용한 해쉬 함수는 다음 성질을 추가로 만족해야 한다.

-collision resistance : h(x)=h(x′)을 만족하는 임의의 서로 다른 두 입력 쌍 x, x′을 발견하는 것이 수행 불가능하다.

거의 대부분의 해쉬 함수의 처리 과정은 입력을 연속적인 고정된 블록들로 나누어 처리함으로서 임의 길이 입력을 해쉬하는 반복적인 처리 과정이다. 먼저 입력 x는 블록 길이의 배수가 되도록 padding 되고 n 개의 블록으로 나누어진다.[1]

해쉬값의 계산은 연쇄 변수에 의존한다. 해쉬 계산을 시작할 때, 이 연쇄 변수는 알고리즘의 일부로 명시된 고정된 초기 값을 가진다. 압축 함수는 해쉬 되어질 메시지 블록을 입력으로 받아 이 연쇄 변수의 값을 갱신하다. 이 과정이 모든 메시지 블록에 대해 순환적으로 반복되고, 연쇄 변수의

마지막 값이 그 메시지에 대한 해쉬값으로 출력된다.

해쉬 함수는 내부 압축 함수로 어떤 구조를 사용하느냐에 따라 3가지로 분류된다. [1]

① 블록암호에 기반한 해쉬 함수
② 모듈러 연산에 기반한 해쉬 함수
③ 전용 해쉬 함수

전용 해쉬 함수는 빠른 처리 속도를 가지고 다른 시스템 서브 요소에 무관하도록 해쉬을 위해 특별히 설계된 함수들이다. 현재까지 제안된 전용 해쉬 함수는 대부분 1990년에 Rivest에 의해 설계된 MD4에 기반한 구조를 가진다. 현재 널리 사용되는 MD 계열 해쉬 함수로 MD5, SHA-1 , RIPEMD-160, HAVAL, HAS160 등이 있다.[2][3]

3. 새로운 구조의 2차원 셀룰라 오토마타

본 연구에서는 새로운 해쉬 함수의 압축 함수로 새로운 구조의 2차원 셀룰라 오토마타를 사용하고 있다.

본 연구에서 개발된 해쉬 함수의 압축 함수로 사용되고 있는 셀룰라 공간 SP는 제안된 해쉬 함수에 사용되는 내부 함수로써 셀룰라 공간 SP의 기하적인 구성은 그림 4와 같이 36개의 사각형들로 분할되어 있는 평면도형으로, 각 셀은 각각 C_0 번부터 C_{35}번까지의 셀 번호를 가지며, 각 셀은 8비트의 입력 값을 가진다.

C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8
C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	C_{16}	C_{17}
C_{18}	C_{19}	C_{20}	C_{21}	C_{22}	C_{23}	C_{24}	C_{25}	C_{26}
C_{27}	C_{28}	C_{29}	C_{30}	C_{31}	C_{32}	C_{33}	C_{34}	C_{35}

그림 1. 셀룰라 공간 SP

동시변환 규칙은 셀룰라 공간 SP의 셀들의 값들을 시각의 변화에 따라 동시에 갱신하는 규칙들을 의미한다. 제안된 해쉬 함수에서는 선형 변환에 기초한 동시변환 규칙 L과 비선형 변환에 기초한 동시변환 규칙 NL이 사용되어 진다.

o 연산 기호의 정의

Xi : i번째 셀의 현재 상태,
Yi : i번재 셀의 다음 상태
a^b : bitwise XOR
a|b : bitwise OR
a<<<n : a의 현재 상태를 정수 n 만큼 왼쪽으로 로테이션
Mi : i번째 32비트 메시지 블록
A3 : 32비트 블럭 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번비트라고 하였을 때, 24번 비트에서 31번 비트까지의 8비트
A2 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번비트라고 하였을 때, 16번 비트에서 23번 비트까지의 8비트
A1 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번비트라고 하였을 때, 8번 비트에서 15번 비트까지의 8비트
A0 : 32비트 블록 A의 오른쪽 끝을 0번으로 A의 왼쪽 끝을 31번비트라고 하였을 때, 0번 비트에서 7번 비트까지의 8비트

o 동시변환 규칙 L

동시변환 규칙 L은 셀룰라 공간 SP의 셀들의 값들을 시각의 변화에 따라 동시에 갱신하는 규칙들 중의 하나로, 각 셀의 다음 상태가 자기 자신과 자신의 왼쪽셀과 자신의 오른쪽 셀의 XOR한 값으로 갱신되는 선형규칙이다. 이때 그림 4의 C_0 , C_9 , C_{18} , C_{27} 셀의 왼쪽 이웃은 각각 C_8 , C_{17} , C_{26} , C_{35} 번 셀로 정의하고, 그림 4의 C_8 , C_{17} , C_{26} , C_{35} 번 셀의 오른쪽 이웃은 각각 C_0 , C_9 , C_{18} , C_{27} 번 셀로 정의한다.

o 동시변환 규칙 NL

동시변환 규칙 NL은 셀룰라 공간 SP의 셀들의 값들을 시각의 변화에 따라 동시에 갱신하는 규칙들 중의 하나로, 각 셀의 다음 상태가 자기 자신과 자신의 왼쪽 셀과 자신의 오른쪽 셀의 XOR한 값과 자신과 자신의 오른쪽 셀의 보수를 XOR한 값

을 다시 OR 한 값으로 갱신되는 비선형규칙이다. 이때 그림 4의 C_0 , C_9 , C_{18} , C_{27} 셀의 왼쪽 이웃은 각각 C_8 , C_{17} , C_{26} , C_{35} 번 셀로 정의하고, 그림 4의 C_8 , C_{17} , C_{26} , C_{35} 번 셀의 오른쪽 이웃은 각각 C_0 , C_9 , C_{18} , C_{27} 번 셀로 정의한다.

5. 제안된 해쉬 함수

본 연구에서 제안된 해쉬 함수는 padding 된 512(32×16)비트의 메시지 블록을 각각 32비트의 16개의 word M[i] (i=0...15)로 나눈 값을 입력으로 받아 각각 32비트인 5개의 해쉬 블록 hj(j=0...5)를 출력으로 하며 160비트의 사용자 키를 사용하여 아래와 같은 단계를 거친다.

단계 1

A←0x67452301, B←0xefcdab89,
C←0x98badcfe, D←0x10325476,
E←0xc3d2e1f0

단계 2

AA←A, BB←B, CC←C, DD←D, EE←E
i ← 0 , j ← 0

단계 3

C_0←AA_3 , C_1←$M[i]_3$, C_2←BB_3 ,
C_3←$K[j]_3$, C4←CC_3 , C_5←$K[j+1]_3$,
C_6←DD_3 , C_7←$M[i+1]_3$, C_8←EE_3 ,
C_9← AA_2 , C_{10}←$M[i]_2$, C_{11}←BB_2 ,
C_{12}←$K[j]_2$, C_{13}←CC_2 , C_{14}←$K[j+1]_2$,
C_{15}←DD_2 , C_{16}←$M[i+1]_2$, C_{17}←EE_2 ,
C_{18}←AA_1 , C_{19}←$M[i]_1$, C_{20}←BB_1 ,
C_{21}←$K[j]_1$, C_{22}←CC_1 , C_{23}←$K[j+1]_1$,
C_{24}←DD_1 , C_{25}←$M[i+1]_1$, C_{26}←EE_1 ,
C_{27}←AA_0 , C_{28}←$M[i]_0$, C_{29}←BB_0 ,
C_{30}←$K[j]_0$, C_{31}←CC_0 , C_{32}←$K[j+1]_0$,
C_{33}←DD_0 , C_{34}←$M[i+1]_0$, C_{35}←EE_0

단계 4

동시변환 규칙 L에 의하여 CA를 1회 갱신

단계 5

C_2 , C_{12} , C_{21} , C_{30}를 연접하여 왼쪽으로 1비트 로테이션

단계 6

C_4 , C_{13} , C_{22} , C_{31}를 연접하여 왼쪽으로 7비트 로테이션

단계 7

C_6 , C_{15} , C_{24} , C_{33}을 연접하여 왼쪽으로 13비트 로테이션

단계 8

단계 4에서 단계7까지를 w2 반복 시행

단계 9

동시변환 규칙 NL에 의해 CA를 1회 갱신

단계 10

단계 5에서 단계7까지를 1회 시행

단계 11

단계 9에서 단계 10을 총 2회 반복 시행

단계 12

M[i]와 M[i+1]을 각각 왼쪽으로 16비트, 왼쪽으로 24비트 로테이션한 값으로 갱신한 결과를 아래와 같이 셀룰라 공간 SP에 입력

C_1 ← $M[i]_3$, C_7 ← $M[i+1]_3$,
C_{10} ← $M[i]_2$, C_{16} ← $M[i+1]_2$,
C_{19} ← $M[i]_1$, C_{25} ← $M[i+1]_1$,
C_{28} ← $M[i]_0$, C_{34} ← $M[i+1]_0$.

단계 13

단계4에서 단계11까지를 1회 반복 시행

단계 14

i ←i+2 , j←j+1

단계 15

단계4에서 단계 14까지를 i=16이 될 때까지 반복

단계 16

AA← C_0 ‖ C_9 ‖ C_{18} ‖ C_{27}
BB← C_2 ‖ C_{11} ‖ C_{20} ‖ C_{29}

$CC \leftarrow C_4 \parallel C_{13} \parallel C_{21} \parallel C_{30}$
$DD \leftarrow C_6 \parallel C_{15} \parallel C_{24} \parallel C_{33}$
$EE \leftarrow C_8 \parallel C_{17} \parallel C_{26} \parallel C_{25}$

단계 17
AA, BB, CC, DD, EE를 해쉬 값으로 출력

단계 18 끝

최초의 512비트 메시지 블록을 처리할 때는 A=0x9e3779b9 , B=0x3c6ef373, C=0x78dde6e6 , D=0xf1bbcdcc , E=0xe3779b99 으로 초기화 된 값을 사용하며, 다음 512비트 블록을 처리할 때는 앞 단계의 해쉬 출력 값을 사용한다.

6. 제안된 해쉬 함수의 검정

제안된 해쉬 함수에 의해서 생성되는 해쉬값의 생성 속도와 해쉬값의 특성에 대한 여러 가지 검정을 다룬다.

Visual C^{++} 6.0으로 구현된 제안된 해쉬 함수의 해쉬 값 생성 속도는 미국의 표준인 SHA-1 수준이었으며 구현된 소스코드의 최적화가 이루어지지 않았음을 고려해볼 때 소스코드의 최적화에 의해 더욱 향상된 속도를 보일 수 있을 것으로 예측된다. 또한 암호학적 해쉬 함수는 주어진 해쉬값에 대해서 이를 만족하는 입력을 찾는 것이 계산적으로 불가능하다는 일방향성과 주어진 입력쌍에 대해 이를 만족하는 임의의 입력 메시지를 찾는 것이 계산적으로 불가능하다는 강한 충돌 회피성을 만족해야 한다. 일방향성은 해쉬 함수에 사용된 라운드 함수의 구조를 파악함으로써 검증가능하며 강한 충돌 회피성의 경우 입력의 변화에 대한 해쉬값을 파악함으로써 검증 가능하다. 이상적인 해쉬 함수에 대하여 512비트의 입력으로부터 160비트의 출력을 생성함으로 서로 다른 입력에 대하여 동일한 출력값을 가질 확률은 2^{352} 이다. 따라서 효율성을 고려해 볼 때 서로 다른 입력쌍에 대하여 동일한 출력값을 가질 확률이 $2^{352}\pm\alpha$ 일 때 강한 충돌회피성을 가지는 것으로 간주한다.

제안된 해쉬 함수의 경우 입력 메시지와 사용자의 키를 혼합하는 라운드 함수로 제안된 셀룰라 오토마타를 이용하고 있으며 제안된 셀룰라 오토마타의 상태전이 함수는 암호학적으로 널리 사용되고 있는 XOR와 OR 그리고 AND 연산자를 사용하고 있으며 선형적인 함수와 비선형적인 함수를 교대로 사용함으로써 일방향성을 만족하고 있다. 또한 제안된 해쉬 함수의 경우 서로 다른 입력쌍에 대하여 동일한 출력값을 가질 확률을 실험에 의하여 계산해 본 결과 약 2^{350} 정도가 되었으며 따라서 제안된 해쉬 함수는 강한 충돌회피성을 가지는 것으로 볼 수 있다.

참 고 문 헌

[1] 이경현, "비밀성 메카니즘 분석 및 평가 기법 연구", 한국정보보호센터, 1998.

[2] 신상욱, 윤재우, 이경현 "셀룰러 오토마타에 기반한 안전한 해쉬함수" 한국멀티미디어논문지 1998,12

[3] A.J. Menezes, P.C. van Oorschot, S.A. Vanstone, Handbook of Applied Cryptography, CRC Press, 1997

[4] S. Wolfram, "Random sequence geration by cellular automata", Advances in Applied Math. 7, 123-169, 1986.

[5] S. Wolfram, "Cryptography with cellular automata", Advances in Cryptology: Proceedings of CRYPTO '85, Lecture Notes in Computer Science 218, 429-432, 1986.

[6] P. Chaudhuri, D. Chowdhury, S. Nandi and S. Chattopadhyay, "Additive cellular automata theory and applications", IEEE Computer Society Press, Los Alamitos, California, 1997.