심층신경망을 이용한 가변 해시 함수 생성기

문성수, 박태준*
전남대학교(학부생), *전남대학교(교수)
dalcw@jnu.ac.kr, *taejune.park@jnu.ac.kr

Variable Hash Function Generator Using DNN

Seongsu Moon, Taejune Park*
Chonnam National Univ., *Chonnam National Univ.

요 약

해시 함수는 보안과 밀접한 연관이 있으므로 충분한 안전성이 보장되어야 하지만 레인보우 테이블(Rainbow Table)을 이용한 공격이나 해시 충돌과 같은 문제점들이 존재한다. 해시 함수가 가지는 이러한 문제들은 해시 함수가 사용되는 시스템 자체를 붕괴시킬 수 있다. 문제를 해결하기 위해서는 모든 해시 함수 객체를 다르게 동작하도록 설계하고 뿐만 아니라 다이제스트(Digest)의 길이를 시스템마다 다르게 하여 다이제스트가 충분한 분산을 가질 수 있도록 하는 가변적인 해시 함수가 적용된다면 해결할 수 있다. 따라서 본 논문에서는 기존 해시 함수가 가지는 문제점들의 원인을 기존 해시 함수의 동작 알고리즘이 고정되어 있다는 것으로 특정하고 이를 해결하기 위해 심층신경망의 특성을 이용하여 유동적으로 동작 알고리즘 및 다이제스트의 길이를 객체마다 다르게 결정할 수 있는 해시 함수를 개발하여 문제들을 해결하고자 한다.

Ⅰ. 서 론

해시 함수는 임의의 길이의 데이터를 입력으로 받아 특정한 길이의 값으로 출력해주는 단방향 암호로써, 오늘날 전자서명, 전자봉투, 전자화폐 등과 같은 여러 분야에서 사용된다 [1]. 이처럼 해시 함수는 보안을 필요로 하는 다양한 분야에서 사용되지만, 평문과 다이제스트의 대응인 레인보우 테이블을 이용한 공격이나, 서로 다른 두 개의 입력값에 대해 같은 다이제스트를 출력하는 해시 충돌과 같은 치명적인 문제가 있다 [1,2]. 해시 함수가 갖는 이러한 문제들의 원인 중 한 가지는 기존 해시 함수가 다이제스트를 생성하는 과정이 특정 연산 알고리즘에 고정되었다는 것에 있다. 따라서 본 논문에서는 이와 같은 문제를 해결하기 위해 해시 함수 자체를 가변적으로 생성할 수 있는 **"심층신경망을 이용한 가변 해시 함수 생성기"**를 제안하고자 한다.

Ⅱ. 본론

2-1 기존 해시 함수의 문제점

(1) 레인보우 테이블을 이용한 공격

레인보우 테이블은 평문과 다이제스트의 대응으로 채워진 거대한 테이블이며 테이블에서 다이제스트의 비교를 통해 평문을 알아낼 수 있다 [2]. 대표적인 해시 함수(MD 계열, SHA 계열 등)들은 방대한 크기의 해시 테이블을 구축하고 있는데 외부 공격자는 해당 해시 함수의 테이블을 이용해 암호문을 평문으로 해독할 수 있다. 따라서 테이블에 아무런 추가 작업 없이 순수 다이제스트만을 저장하는 것은 비교적 쉽게 평문을 알아낼 수 있으므로 위험하다.

(2) 해시 충돌

해시 함수의 다이제스트가 표현할 수 있는 범위는 매우 넓지만, 무한대는 아니므로 우연히 다른 입력에 대해서 같은 다이제스트가 출력되는 현상인 해시 충돌이 발생할 수 있다 [1]. 해시 충돌이 발생한다면 운영체제와 같은 시스템에서 예상치 못한 사용자가 권한을 획득하는 불상사가 발생할 수 있다 [3].

2-2 제안하는 해시 함수

본 논문에서는 해시 함수가 다이제스트를 도출하는 내부 동작뿐만 아니라, 다이제스트의 길이까지 조정 가능한 "변화 가능한 해시 함수"를 통해 위에서 언급한 문제들을 해결하고자 한다. 이처럼 가변적이면서도 각 해시 함수가 고유한 성질을 갖도록 하기 위해서 학습 데이터에 따라 동작을 가변적으로 변경할 수 있으며, 모델이 출력하는 결과의 길이를 임의로 조절할 수 있는 심층신경망을 이용하여 변화 가능한 해시 함수를 구현하려고 한다 [4]. 심층신경망으로 구성된 각 해시 함수 객체는 학습한 데이터에 따라 모두 다르게 동작하므로 레인보우 테이블의 생성 자체가 무의미할 것이며, 다이제스트의 길이를 조절함으로써 다이제스트의 분포를 늘려 자연스럽게 해시 충돌의 문제를 완화할 수 있다.

2-3 심층신경망을 이용한 해시 함수 구현

(1) 개요

시스템은 문자(Character) 단위로 입력을 받고 이에 맞는 특징 벡터를 구하는 해시 블럭(Hash Block)모델과 해시 블럭으로부터 추출된 특징 벡터를 통합하는 합병 블럭(Merge Block)이라는 두 개의 심층신경망 모델로 구성된다. 이 시스템의 핵심 동작 원리는 함수의 출력이 다음 함수의 입력으로 들어가는 구조인 SHA-256모델에서 착안하였다 [5,6]. 이하 본 논문에서는 설명상 편의를 위해 다이제스트의 길이를 64비트로 고정하지만, 모델이 출력하는 값의 길이를 조절함으로써 다이제스트의 길이는 가변적으로 설정할 수 있다.

(2) 시스템의 학습 데이터 구축

일반적인 심층신경망 모델은 데이터의 패턴을 인식시키기 위해 학습을 하지만, 본 시스템은 입력에 따른 출력값이 예측 불가능해야 하므로 모델이 특정 패턴을 학습해서는 안 된다. 따라서 아무런 의미가 없는 랜덤 데이터를 이용하여 모델을 학습하여 특정 패턴을 갖지 않도록 하려고 한다. 즉, 랜덤으로 생성된 학습 데이터는 오직 파라미터 조정 및 해시 함수의 동작 방식 결정에만 사용한다.

(i) 해시 블럭 모델 학습 데이터

해시 블럭은 한 개의 문자를 입력받아 최종 해시 함수의 다이제스트 길이와 같은 길이의 특징 벡터를 추출하는 모델이다. 따라서 본 모델을 학습하기 위한 학습 데이터의 입력은 한 개 정수이며, 이에 따른 정답 데이터는 다이제스트 길이와 같은 벡터가 되도록 랜덤으로 구성한다.

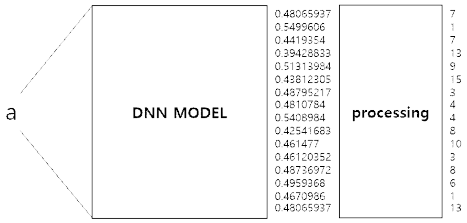
(ii) 합병 블럭 모델 학습 데이터

합병 블럭은 해시 블럭으로부터 문자 단위로 추출한 특징 벡터들을 통합하여 최종 다이제스트를 만드는 모델이다. 특징 벡터들을 통합하는 방식에는 여러 가지가 있을 수 있지만 본 논문에서는 두 개의 특징 벡터를 입력받아 이들을 통합하여 다이제스트의 길이와 같은 길이의 특징 벡터를 출력하도록 설계하였다. 따라서 합병 블럭에서 학습 데이터의 입력은 다이제스트의 길이의 두 배인 랜덤으로 구성된 벡터이며, 정답 데이터는 다이제스트의 길이와 같은 벡터가 되도록 랜덤으로 구성한다.

(3) 모델 구축

(i) 해시 블럭

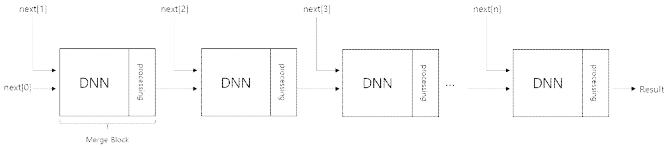
[그림 1]과 같이 해시 블럭은 한 개의 문자를 입력으로 받아 0 ~ 15 사이의 값 16개를 가지는 벡터를 추출하도록 완전 연결 층(Fully Connected Layer)을 이용하여 구축한 후 이전 단계에서 생성한 해시 블럭용 랜덤 데이터를 이용하여 모델을 학습한다. 해시 함수는 수치상 인접한 입력이라도 완전히 다른 결과가 출력되어야 하지만 학습된 모델에게 수치상 인접한 값, 예를 들어 'A'(0x41)와 'B'(0x42)를 입력으로 한 후 출력된 벡터의 동일한 위치에 있는 임의의 원소를 살펴본다면, 각각 0.503987014293671, 0.503930687904358과 같이 0.5039와 같은 높은 자리의 출력값이 유사하다는 문제가 발생한다. 그러나 예시에서 0.5039 이하의 충분히 작은 값은 인접한 입력이라 하더라도 크게 차이가 나기 때문에 큰 변화가 발생하는 부분까지를 정수로 취하여 문제를 해결할 수 있다. 이후 모듈러(Modular) 연산을 이용하여 각 원소의 값의 범위를 0 ~ 15 사이의 정수로 조정하여 최종 특징 벡터를 도출한다. 물론 입력에 대해 출력하는 값은 학습된 객체마다 다르다.



[그림 1] 해시 블럭 모듈 예제

(ii) 합병 블럭

합병 블럭은 문자 단위로 생성된 특징 벡터들을 합병하도록 완전 연결 층을 이용하여 설계된 모델이며 이전 단계에서 생성한 합병 블럭용 랜덤 데이터를 이용하여 학습한다. 합병 시스템의 전반적인 과정은 [그림 2]와 같이 가장 첫 단계에서는 해시 블럭을 통과하여 나온 특징 벡터 두 개를 입력으로 받아 다이제스트의 길이만큼의 또 다른 벡터를 출력한다. 첫 단계 이후 과정부터는 이전 단계를 통과한 값과 아직 합병되지 않은 특징 벡터를 동시에 다음 합병 블럭에 통과 시킨다. 여기서 이전 단계의 모델이 출력한 값을 바로 다음 단계의 모델 입력의 일부로 들어가는 것이 아니라, 0 ~ 15 사이의 정수들로 구성된 특징 벡터들과의 형태 일치 및 최종 다이제스트의 형태를 위하여 모델 출력값을 0 ~ 15 사이의 정수로 단계마다 조정한다. 이와 같은 과정은 특징 벡터들이 한 개의 벡터로 통합될 때까지 반복된다. 최종적으로 생성된 한 개의 벡터는 해시 함수의 다이제스트가 된다.



[그림 2] 합병 블럭 모듈 예제

(4) 결과

구현된 해시 함수에 대해 구글 코랩(Google Colaboratory) Tesla T4 GPU 환경에서 실험을 진행하였다. 본 해시 함수에 유사하게 생긴 단어 "Preserve"와 "Persevere"를 입력으로 주었을 때 각각 "191807a73de9ff79", "c4e0b2593beb492d"가 결과로 출력되었고 외관상 비슷한 단어이지만 64비트 길이의 서로 다른 다이제스트를 출력한다는 것을 확인하였다.

Ⅲ. 결론

본 논문에서는 기존 해시 함수의 문제들을 해결하기 위해 심층신경망을 이용하여 객체마다 다른 동작을 하는 가변 해시 함수를 고안하였다. 가변 해시 함수를 구성하는 심층신경망 모델을 임의의 데이터로 학습시켜 각 학습된 모델 객체마다 같은 평문이라도 다른 다이제스트를 도출하도록 하여 유일한 해시 함수를 만들 수 있다는 것을 보였다. 이러한 가변 해시 함수는 기존의 해시 함수가 가지는 문제점을 개선할 뿐만 아니라 외부 공격으로부터 해시 함수 시스템에 문제 발생 시 해시 함수 자체를 빠른 시간내에 변경할 수 있어 신속한 보안 문제 해결에 도움을 줄 수 있으며, 즉시 해시 함수를 생성할 수 있다는 장점을 살려 일회성 검증 시스템과 같은 분야에서 유용하게 사용될 것으로 사료된다.

향후 연구로는 가변 해시 함수는 각각의 객체마다 동작 방식이 다르다는 점에 유의하여 방대한 데이터를 입력으로 통과시켜 다이제스트의 분포를 확인하는 기존 해시 함수 검증 체계에서 벗어나 빠른 시간 내에 안전성을 확인할 수 있는 검증 시스템에 대해 연구하고자 한다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2022R1C1C1006967).

참 고 문 헌

[1] 홍남수, 강정호, 전문석. (2017). 해시 알고리즘의 안전성 동향. 한국정보처리학회 학술대회논문집, 24(1), 459-460.

[2] H. Kumar et al., "Rainbow table to crack password using MD5 hashing algorithm," 2013 IEEE Conference on Information & Communication Technologies, 2013, pp. 433-439, doi: 10.1109/CICT.2013.6558135.

[3] 배유미, 정성재, 소우영. (2016). 리눅스에 적용된 해시 및 암호화 알고리즘 분석. 한국항행학회논문지, 20(1), 72-78.

[4] 이재성. (2016). 심층 신경망의 발전 과정과 이해. 한국통신학회지(정보와 통신), 33(10), 40-48

[5] 이상현, 신경욱. (2018). IoT 보안을 위한 SHA-256 해시 프로세서의 면적 효율적인 설계. 한국정보통신학회논문지, 22(1), 109-116.

[6] 한국인터넷진흥원(KISA). 암호알고리즘 소스코드. KISA_SHA256 소스코드https://seed.kisa.or.kr/kisa/Board/21/detailView.do