한국컴퓨터정보학회 하계학술대회 논문집 제26권 제2호 (2018. 7)

스크래치 기반의 암호화 프로그램

허태성O, 이민재*, 김가겸*
O*인하공업전문대학 컴퓨터정보과
e-mail: tshur@inhatc.ac.kr, aa53420@gmail.com, rkrua5283@gmail.com

Encryption Program using Scratch

Tai-Sung HurO, Min-Jae Lee*, Ga-Gyeom Kim*
O*Dept. of Computer Science, Inha Technical College

● 요 약 ●

일반적으로 정보를 전달함에 있어 정보의 유출은 큰 문제이다. 정보를 전달하는 방법이 발달하고 보편화됨에 따라 오늘날에 와서는 개인정보 유출과 관련된 문제가 지속적으로 대두되었다. 개인정보의 보호가 더욱 중요하게 생각되는 현 상황을 고려하여 스크래치(Scratch)의 기본 연산기능을 이용한 한글과 특수문자, 영어 암호화(Encryption) 및 복호화(Decryption)를 가능하게 하고, 정수 형태의 2개의 개인키와 간단한 알고리즘을 통해 암호문을 생성하는 어플리케이션을 통해 암호화와 복호화에 대한 개념을 학습하고, 더욱 나아가 개인정보 보호에 대한 중요성을 상기할 수 있도록 하였다.

키워드: Scratch, Encryption, Decryption

I. 서 론

일반적으로 정보를 전달하는 것은 매우 중요한 일이며, 그에 따라 정보를 전달하는 방법은 계속 발달해왔고 발전된 정보 전달 기술들이 보편화되면서 수많은 양의 정보들이 더욱 빠르게 이동하고 있다. 빠르게 정보를 전달하는 것도 중요하지만, 그 내용이 누설된다면 그 정보는 가치를 상실할 것이다. 과거에도 많은 정보의 탈취로 인해 피해들이 발생해왔으며, 이를 막기 위해 정보를 암호화하는 것 또한 같이 발전해왔다. 암호화란, 정보를 탈취하고자 하는 사람이 정보를 훔쳐도 그 내용을 알 수 없도록 하는 것이다. 본 시스템은 간단한 로직을 통해 평문 (Plain Text)을 암호문 (Cypher Text)로 만들며, 암호화 로직을 알아내더라도 그 값을 쉽게 알아내지 못하도록 개인키 개념을 도입하였다. 암호화는 일반적으로 어려운 알고리즘이나 복잡한 수식이 떠오르기 쉽다. 하지만, 본 연구에서는 간단한 1차함수와 개인키의 개념을 통해 쉽게 암호화와 복호화에 대한 개념에 대해 쉽게 접근할 수 있고 실제로 해독하기 어려운 암호문을 만들어보고자 한다.

II. 본 론

1. 1차 함수

1차 함수는 그래프가 직선인 다항 함수이며 다음과 같은 꼴을 가진다.

$$(f(x) = ax + b \ \text{단}, a\text{와 } b\text{는 정수})$$

암호화(Encryption)를 통해 평문(Plain Text)은 암호문(Cypher Text)이 되고, 암호문은 다른 사람들이 보았을 때 이해할 수 없는 문장으로 만들어지게 된다. 암호문을 다시 평문으로 만들기 위해서는 복호화(Decryption) 과정을 거쳐야 한다. 즉, 암호화에 사용되는 함수는 역함수가 존재해야하고, 이 역함수가 복호화 과정을 뜻하게 된다. 이를 위해서는 해의 유일성을 보장해야 한다. 간단한 수식임에도 불구하고 Key가 2개를 이용하여 순서쌍 (a, x, b)의 경우의 수가 무한에 가까운 해를 가지게 된다. 만약 키를 하나만 사용했을 경우 $(y = ax)$ 암호화한 결과가 6일 때, 순서쌍 (a, x)의 경우의 수는 8개다. 물론, 암호화한 결과가 커질수록 경우의 수도 늘어나지만 그 수는 여전히 유한적이다. 컴퓨터의 빠른 연산능력을 사용한다면 이 유한의 경우의 수를 찾는 것은 시간문제이며, 암호화 로직은 순식간에 해석된다. 이에 따른 문제를 해결하기 위해 Key를 2개를 사용하게 되었다.

2. 한글, 공백, 영어, 특수문자 변환

앞서 소개한 함수를 이용하여 평문을 암호문으로 변환하기 위해서는 입력된 한글, 공백, 영어, 특수문자를 숫자로 변환해야한다. 공백, 영어, 특수문자는 ASCII 코드표를 통해 숫자로 변환 하였다. 한글은 별도의 리스트를 통해 영문으로 변환하고 그 영문을 ASCII코드에 매칭되는 숫자로 변환한다.

들어온 값이 영문일 경우

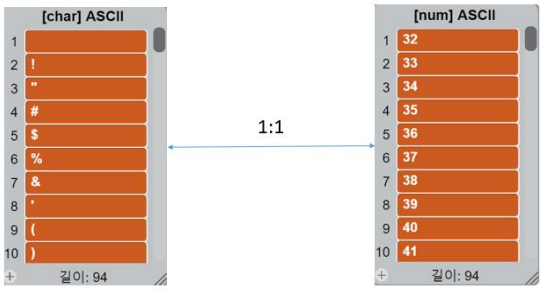

Fig. 1. 영문값 변환

들어온 값이 한글일 경우

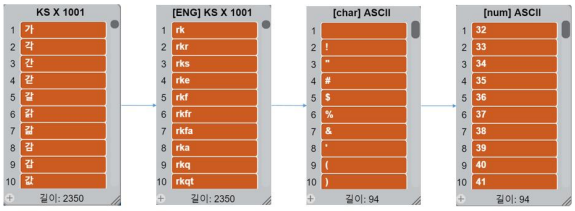

Fig. 2. 한글값 변환

3. 프로그램 실행

Fig. 3. "inhatc 화이팅!" 암호화 결과

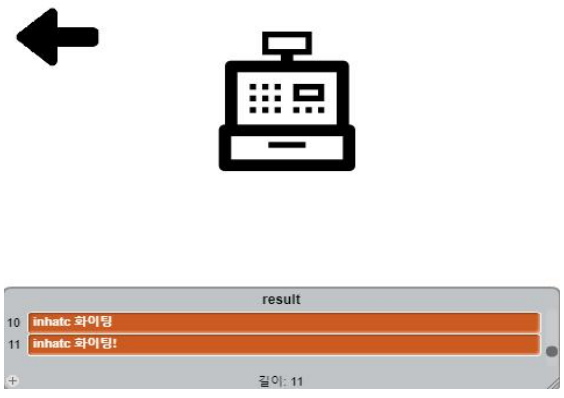

Fig. 4. "inhatc 화이팅!" 복호화 결과

III. 결 론

2개의 개인 Key를 이용한 1차 함수를 통해 무한에 가까운 경우의 수를 가지는 순서쌍을 가진 암호문을 만들 수 있으며, 쉬운 예를 통해 암호학에 대해 어렵다는 고정관념을 환기시킴과 동시에 관심을 유도할 수 있고 나아가 암호화와 복호화의 개념을 직접 학습할 수 있다는 장점이 있다.

REFERENCES

[1] 위키피디아 – 암호화, https://ko.wikipedia.org/wiki/%EC%95%94%ED%98%B8%ED%99%9
[2] 위키피디아 - 암호학, https://ko.wikipedia.org/wiki/%EC%95%94%ED%98%B8%ED%95%99
[3] 위키피디아 - 1차함수
[4] 위키피디아 – 역함수