대화형 AI 기반 RSA 암호화 알고리즘 구현

김동현, 한승주, 원민철, 김규민, 김용강*
국립공주대학교

raehdgus@naver.com, hansengjuu@naver.com, mnmkjmin2896@gmail.com, songkgm@gmail.com,
*ygkim@kongju.ac.kr

Implementation of conversational AI-based RSA encryption algorithm

Donghyeon Kim, Seungju Han, Mincheol Won, Gyumin Kim, Yonggang Kim*
Kongju National University

요 약

본 논문에서는 대화형 artificial intelligence (AI) 기술을 활용한 RSA 암호화 알고리즘을 제안 및 구현한다. 제안한 방법은 speech-to-text (STT)와 text-to-speech (TTS) 기술을 이용하여 사람의 음성을 확인하고, 음성 내 포함되어 있는 명령어 및 비밀번호를 인지하여 Rivest-Shamir-Adleman (RSA) 암호화 및 복호화 과정을 수행한다. 제안한 대화형 AI 기술 기반 RSA 알고리즘을 Java 및 Python 을 이용하여 구현하고 타당성을 검증한다.

Ⅰ. 서론

다양한 노드들의 통신 및 인터넷 서비스 요구량이 증가함에 따라 개인 정보 및 데이터의 보호가 중요해졌다. 정보 보호를 위해 다양한 암호화 기술 및 프로그램이 개발되고 있다 [1]. 하지만, 시각장애인 등 취약 계층에 대한 암호화 프로그램의 접근성이 낮은 문제점이 존재한다. 본 논문에서는 대중성 및 활용성이 높은 Rivest-Shamir-Adleman (RSA) 암호화 방식에 대화형 artificial intelligence (AI) 기술을 이용하여 취약 계층에 대한 접근성을 높일 수 있는 암호화 프로그램을 제안한다. 또한 Java 및 Python 을 이용하여 제안한 암호화 프로그램을 구현하고 실생활 적용을 위한 활용성에 대해 검증한다.

Ⅱ. 대화형 AI 기반 암호화 프로그램 구성 및 구현

II-1 프로그램 구성

본 연구에서 제시한 프로그램은 크게 “대화형 AI 가 적용된 클라이언트”와 “암호화 및 database (DB) 담당 서버”로 구분된다(그림 1). 클라이언트와 서버는 소켓통신으로 구성되어 데이터를 주고 받는다(그림 2). 클라이언트에선 메시지 전송을 위해 두 번의 전송을 하는데, 첫 번째 전송에선 4 바이트로 데이터의 길이를 전송하여, 서버에 그 크기만큼의 배열을 생성한다. 두 번째 전송으로 메시지를 전송해 해당 배열에 저장한다. 서버는 메시지를 받아 서버에 저장한 뒤 대기상태에 들어간다. 이때, 모니터 역할을 할 Boolv 의 Boolean 값을 바꾸고, 이후 반복문 안에서 계속 모니터링한다. 암호화 프로그램에선 서버 내부의 값을 읽은 뒤 Boolv 의 Boolean 값을 원래대로 바꿔준다. 서버 내부에선 바뀐 값을 읽고 반복문을 break 한 후, 클라이언트의 전송을 기다린다. 암호화 및 복호화 데이터는 서버와 연결된 DB 에 저장된다. MySQL 을 사용하여 DB 를 구성하였으며, 암호화 대상 파일 이름과 RSA 알고리즘 동작을 위한 암호화 키가 저장된다.

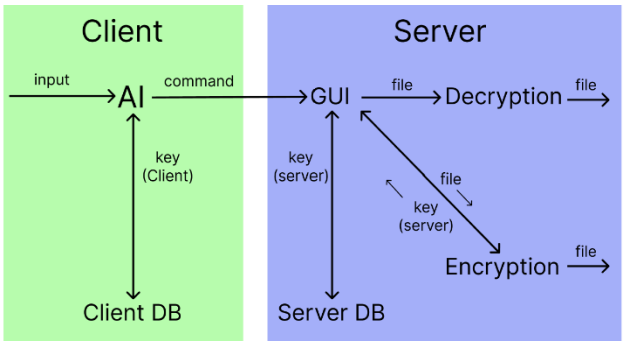

그림 1 대화형 AI 기반 RSA 암호화 프로그램 구성.

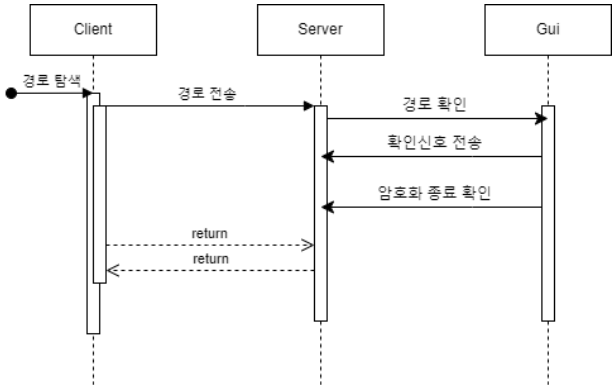

그림 1 클라이언트와 서버 간 통신 플로우.

II-2 암호화 기능

본 프로그램은 랜덤 클래스를 이용해서 랜덤한 숫자를 발생시킨 후 이 숫자를 암호키로 이용해서 파일의 비트와 XOR 연산을 시켜서 파일을 암호화한다. 그 후에 랜덤한 RSA 암호화를 이용하여서 암호키를 암호화시킨 후에 데이터베이스에 저장시킨다.

랜덤키 발생 과정은 다음과 같다. 파일을 읽어서 첫 4 개의 바이트를 가져온다. 그 후에 4 개의 난수를 발생시켜서 4 개의 숫자 중에 1 개의 숫자를 결정시킨다.

이 과정을 2 회 시행해서 2 개의 암호키를 만든 후 byte 배열을 만들어 저장한다. 그리고 임의 수 index 를 0 으로 초기화시키고 파일을 읽으면서 index 를 증가시키고 파일의 바이트와 암호키 배열의 (index % (암호키 배열의 길이))번째의 수를 XOR 연산을 거친 후에 파일이름을 (기존 파일이름+cry)인 파일을 만들어서 저장시킨다. 파일 암호화가 끝난 후에 암호키 배열을 데이터베이스로 반환시켜준다.

암호화키 배열을 암호화해주기 위해서 RSA 암호화키를 생성해야 한다. 난수 생성을 통해서 100~200 사이의 소수들을 발생시켜서 이를 이용하여서 수의 크기가 256 이하인 바이트를 암호화시킬 수 있는 RSA 암호화 공개키, 개인키를 생성하여 준다. 이를 바탕으로 전에 만들었던 암호화키 배열을 암호화하고 이를 데이터베이스에 반환시킨다.

II-3 대화형 AI 기술 적용

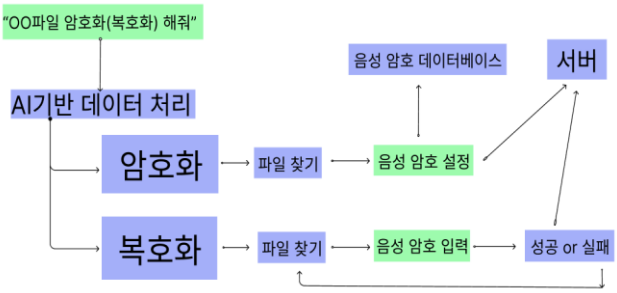

그림 3 제안 프로그램 동작 흐름도

본 프로그램에서는 음성 데이터를 텍스트 데이터로 바꿔주는 speech-to-text (STT)와 텍스트 데이터를 음성 데이터로 바꿔주는 text-to-speech (TTS) 기술을 이용하여 대화형 AI 기반 암호화 프로그램을 구현한다. 사람의 음성으로 명령을 내리고 그 음성 데이터가 STT 를 통해 텍스트 데이터, 즉 컴퓨터가 알아 들을 수 있는 데이터 형태로 바뀐 다음 이벤트를 진행하고 이벤트의 반환 값을 TTS 를 통해 음성으로 바꿔주어 스피커로 출력하도록 한다. 본 프로그램에서는 Google 에서 제공하는 음성 텍스트 변환 API 를 사용하여 프로그램을 구성하였다 [2]. 이벤트가 일어나는 함수에서는 우리가 이루고자 하는 암호화와 복호화를 진행시킨다. 예를 들어 'XX 파일 암호화(복호화) 해 줘' 라는 말이 들어오면 그 값이 텍스트 데이터로 변환된 후 '암호화'라는 말이 들어있다면 암호화를, '복호화'라는 말이 들어있다면 복호화를 진행한다.

암호화 과정은 '암호화'라는 말과 파일명이 들어오게 됐다면 클라이언트에서 파일명 이외에 데이터를 지우는 과정을 거친다. 이를 통해 파일명만을 다른 변수에 넣을 수 있고 이 파일명을 통해 컴퓨터 안에 있는 파일을 찾을 수 있다. 찾은 후에는 그 파일의 절대경로를 반환해 클라이언트와 연결되어있는 서버에 절대경로를 전달한다. 전달된 경로를 인식한 서버는 그 파일을 암호화하게 된다. 서버가 파일을 암호화하게 된 후 클라이언트에서도 또 다른 음성 암호화가 시작된다. 이 암호화를 통해 사용자가 음성 비밀번호를 걸어 다른 누군가가 접근하지 못하도록 하기 위함인데 이 암호는 음성을 통해 인식해 저장된다.

복호화 과정은 다음과 같다. 사용자가 '복호화'라는 말과 파일명을 넣는다면 복호화가 진행된다. 이때 다른 점이라고 함은 파일을 찾고 곧장 복호화가 되는 것이 아닌 파일을 찾은 후 이전에 설정한 비밀번호를 말해야 파일의 절대경로가 넘어가 복호화가 진행된다는 점이다. 만약 암호를 잊었다면 다시 처음으로 돌아간다. 암호가 맞았다면 csv 파일에 있는 파일명과 비밀번호를 확인한 후 Cosine Silmilarity 를 통해 맥락을 파악한 후 일치한다면 복호화를 진행한다.

II-4 프로그램 구현 및 동작 결과

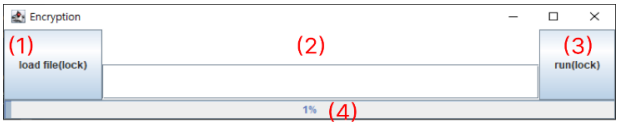

그림 4 암호화 프로그램 동작

구현한 프로그램의 사용자 그래픽 인터페이스 환경은 그림 4 와 같다. (1)에서는 수동 파일을 선택한다. (2)에서는 암호화 및 복호화 대상으로 선택된 파일의 위치 출력 및 오류 발생시 알림을 표시한다. (3)에서는 대화형 AI 외 수동으로 암호화 및 복호화를 실행할 수 있는 기능을 제공한다. (4)에서는 암호화 및 복호화 진행도를 표시한다. 프로그램의 주요한 동작 기능은 다음과 같다.

- 복호화에 필요한 데이터 로드
- 암호화 및 복호화 명령
- 암호화 결과물에 대한 DB 전달 및 관리

DB 에 데이터를 가져오지 못할 경우 현재 파일은 암호화된 적이 없다는 것이므로, 자동적으로 암호화를 처리한다. DB 에 데이터를 가져온다면 반대로 복호화가 처리된다. 이때 통합함수는 결과물에서 차이가 발생하는데 결과물을 비교해 이가 암호화된 것인지, 복호화 된 것인지를 감지하여 암호화된 경우 DB 에 전달하고 데이터를 관리한다.

Ⅲ. 결론 및 향후 연구 방향

본논문에서는 대화형 AI 기반 RSA 암호화 프로그램을 제안 및 구현하였다. 제안한 프로그램은 시각장애인 등 취약계층에 대한 암호화 프로그램 접근성을 향상시킨다. 향후 암호화 프로그램에 적합한 STT 및 TTS 기술 및 음성 분석을 통한 화자 인식 기술을 적용하여 프로그램 성능을 향상시키고자 한다.

ACKNOWLEDGMENT

이 성과는 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. RS-2022-00166739).

참 고 문 헌

[1] E. Abouelkheir and S. El-Sherbiny, "Enhancement of speech encryption/decryption process using RSA algorithm variants," *Human-centric Computing and Information Sciences*, vol. 12, 2022.

[2] Google Speech-to-Text, https://cloud.google.com/speech-to-text/docs/