한국사물인터넷학회논문지 Vol. 2, No. 3, pp. 43-51, 2016 http://dx.doi.org/10.20465/KIOTS.2016.2.3.043

Crypft+ : Python/PyQt 기반 AES와 HASH 알고리즘을 이용한 파일 암복호화 시스템

신동호, 배우리, 신형규, 남승진, 이형우*
한신대학교 컴퓨터공학부

Cryptft+ : Python/Pyqt based File Encryption & Decryption System Using AES and HASH Algorithm

Dongho Shin, Woori Bae, Hyeonggyu Shin, Seungjin Nam, Hyung-Woo Lee*
Division of Computer Engineering, Hanshin University

요 약 본 논문에서는 IoT 시스템 또는 개인별 문서 파일 관리 과정의 보안성을 향상시키기 위해 개선된 파일 암복호화 시스템인 Crypft+를 개발했다. Crypft+ 시스템은 Python을 이용하여 핵심 보안 모듈을 개발하였으며, PyQt를 사용하여 사용자 인터페이스를 설계 및 구현하였다. 또한 가장 보안성이 뛰어난 AES 기반 대칭키 암호 알고리즘과 SHA-512 기반 해쉬 알고리즘을 이용하여 컴퓨터 시스템 내부에 저장된 중요 파일에 대한 암호화 및 복호화 과정을 수행할 수 있도록 구현하였다. 또한 Cx-Freezes 모듈을 사용하여 구축된 프로그램을 exe 기반 실행 파일로 변환하는 기능을 구현하였으며, 프로그램 사용에 있어 이해를 돕는 설명서를 프로그램 내부에 포함시켜 직접 다운로드 받을 수 있도록 구현하였다.

주제어 : 파이썬 PyQt, 보안 파일 관리 시스템, 암복호화 시스템.

Abstract In this paper, we have developed Crypft+ as an enhanced file encryption/decryption system to improve the security of IoT system or individual document file management process. The Crypft+ system was developed as a core security module using Python, and designed and implemented a user interface using PyQt. We also implemented encryption and decryption function of important files stored in the computer system using AES based symmetric key encryption algorithm and SHA-512 based hash algorithm. In addition, Cx-Freezes module is used to convert the program as an exe-based executable code. Additionally, the manual for understanding the Cryptft+ SW is included in the internal program so that it can be downloaded directly.

Key Words : Python, PyQt, Secure File Management System, Encryption & Decryption System.

1. 서론

최근 보안에 대해서 관심이 높아지고 있는 추세이다. 따라서 대기업의 경우 자체 개발로 만든 암복호화 프로그램을 사용하여 파일 교환 시 자체 개발한 파일 암호화 모듈을 적용하고 있다. 하지만 중소기업과 개인의 경우

본 논문은 한신대학교 2016년도 2학기 컴퓨터공학부 종합설계2 프로젝트 결과물입니다.
*교신저자 : 이형우(hyungwoo8299@gmail.com)
접수일 2016년 10월 2일 심사완료 2016년 10월 28일

별다른 보안 모듈 적용/보안 조치 없이 파일을 송수신하므로 보안상의 문제점이 발생한다.

중소기업과 개인의 경우 파일 암복호화 프로그램을 개발하기에는 많은 예산과 시간이 소요된다. 때문에 업무 상 중요 파일에 대한 송수신 시 보안성을 향상시킬 수 있는 암복호화 핵심 모듈 개발이 필요하다.

본 논문에서는 중소기업과 개인의 문서 파일 보안성을 제공하기 위해 파일 암복호화 핵심 엔진인 Crypft+ 시스템을 개발했다. Python을 이용하여 주요 기능을 개발하였으며, Pyqt 기반으로 제작하였다. 가장 보안성이 높은 AES 암호 알고리즘을 이용하여 파일을 암복호화 할 수 있다.

프로젝트는 암호화, 복호화, HASH 세 가지 기능으로 구성된다.

암호화 기능에서 파일 선택과 파일 암호화에 사용할 비밀번호를 결정하고, 생성될 파일의 위치와 이름을 수정할 수 있으며, 파일 정보를 암호화된 데이터와 함께 생성한다.

복호화 기능도 암호화 기능과 동일하게 파일 선택 및 비밀번호를 작성하고, 암호화 과정에서 입력된 파일 정보로 비밀번호 검사와 원본 파일에 대한 무결성을 확인한다.

HASH 기능은 선택한 파일을 읽어서 해시 값을 확인한다.

본 논문의 구성은 다음과 같다. 2장에서는 논문에 사용된 관련기술에 대하여 서술하고, 3장에서는 설계 및 구조의 특성을 작성했다. 4장에서는 Crypft+의 구현 내용을 기술하고 있다. 5장에서는 결론을 제시했다.

2. 관련 기술

2.1 Python

Python은 초보자부터 전문가까지 사용자층을 보유하고 있다. 동적 타이핑(dynamic typing) 범용 프로그래밍 언어로, 펄 및 루비와 자주 비교된다. 다양한 플랫폼에서 쓸 수 있고, 라이브러리(모듈)가 풍부하여, 대학을 비롯한 여러 교육 기관, 연구 기관 및 산업계에서 이용이 증가하고 있다. 또 Python은 순수한 프로그램 언어로서의 기능 외에도 다른 언어로 쓰인 모듈들을 연결하는 풀 언어(glue language)로써 자주 이용된다.

[그림 2.1] Python Logo

실제 Python은 상용된 많은 응용 프로그램에서 스크립트 언어로 채용되고 있다. 도움말 문서 정리가 잘 되어 있으며, 유니코드 문자열을 지원해서 다양한 언어의 문자 처리에도 능하다. Python은 기본적으로 해석기(인터프리터) 위에서 실행될 것을 염두에 두고 설계되었다.

2.2 Pyqt

Pyqt 는 영국의 Riverbank Computing 이라는 곳에서 C++ 의 Cross Plaform GUI Framework 중 하나인 Qt를 Python 모듈로 변환해 주는 툴을 만들며 시작되었다. Qt는 본래 Python에서 사용할 수 없는 C++용이지만, Python에서도 사용할 수 있게 변환한 툴이 Pyqt이다. Pyqt인 경우 Python과 Qt를 동시에 사용해야 한다.

2.3 Cx-Freezes

Cx-Freezes는 py2exe와 py2app 같은 방식으로 실행할 수 있는 Python 스크립트로서 파일을 제작하는 모듈이다. 위의 두 가지 도구와는 달리 Cx-Freezes는 Python 자체가 작동하는 모든 플랫폼이고, 크로스 플랫폼이다. Python 3과 Python 2.4 이상 버전이 필요하다.

2.4 AES 암호 알고리즘

고급 암호화 표준(AES, Advanced Encryption Standard)은 2001년 미국 표준 기술 연구소(NIST)에 의해 제정된 암호화 방식이다. AES는 두 명의 벨기에 암호학자에 의해 개발된 Rijndael 암호에 기반하며 AES 공모전에서 선정되었다.

미국 표준 기술 연구소(NIST)는 2001년 11월 26일 AES를 미국 연방 정보 처리 표준(FIPS-197)으로 공포하였다. NIST는 5년의 표준화 과정을 거쳤으며 이 과정에서 15개의 알고리즘이 경쟁, Rijndael 암호가 가장 적합한 알고리즘으로 선정되었다. 이 표준은 2002년 5월 26일부터 효력을 발휘하기 시작했다. AES는 ISO/IEC 18033-3 표준에 포함되어 있으며 여러 암호화 패키지에서 사용되고 있다.

2.5 HASH

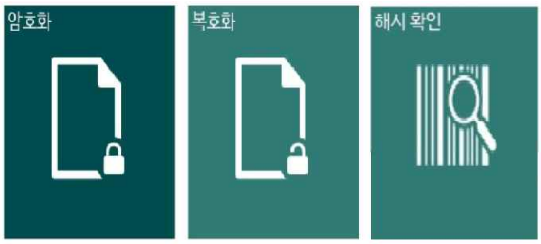
해시 함수는 임의의 길이의 데이터를 고정된 길이의

데이터로 매핑(mapping)하는 함수이다. 해시 함수에 의해 얻어지는 값은 해시값, 해시 코드, 해시 체크섬(checksum) 또는 간단하게 해시라고 한다. 그 용도 중 하나는 해시 테이블이라는 자료구조에 사용되며, 매우 빠른 데이터 검색을 위한 컴퓨터 소프트웨어에 널리 사용된다. 해시 함수는 큰 파일에서 중복되는 레코드를 찾을 수 있기 때문에 데이터베이스 검색이나 테이블 검색의 속도를 가속할 수 있다. 암호용 해시 함수는 매핑된 해싱값만으로는 원래 입력 값을 알아내기 힘들다는 장점에 의해 사용될 수 있다. 또한 전송된 데이터의 무결성을 확인해주는 데 사용되기도 하는데, 메시지가 누구에게서 온 것인지 입증해주는 HMAC를 구성하는 블록으로 사용된다.

3. 설계 및 구조

3.1 시스템의 기능

시스템에 구현된 내용은 암호화, 복호화, HASH 세 가지 기능으로 구성된다.



[그림 3.1] 시스템에 구현된 기능

암호화 기능은 파일 선택, 비밀번호 설정, 경로 설정, 미리보기 및 검토, 진행, 완료 페이지로 구성된다. 파일 선택 페이지에서는 파일의 이름과 파일 용량, 파일 확장자, 파일 경로를 출력한다. 파일 비밀번호 설정 페이지에서는 사용자 입력의 비밀번호 설정, 또는 임의 값을 비밀번호로 지정하는 것이 가능하다. 임의 값을 비밀번호로 설정할 때는 별도의 파일에 비밀번호를 저장하고, 경로를 설정할 수 있다. 경로 설정 페이지에서는 암호화 하여 생성될 파일의 경로와 생성될 파일의 이름을 수정할 수 있다.

미리보기 및 검토 페이지에서는 파일 명, 파일의 생성 경로, 생성될 암호 경로, 용량 정보를 검토할 수 있다. 진행 페이지에서는 스레드를 사용하여 시간과 암호화 과정을 동시에 표시한다. 마지막으로 완료 페이지에서는 생성된 암호화 파일의 위치를 열어주는 항목을 삽입했다. 프로그램 이름, 확장자, 비밀번호 해시값, 원본 파일의 해시값, 데이터 순으로 암호화 파일을 생성 시킨다.

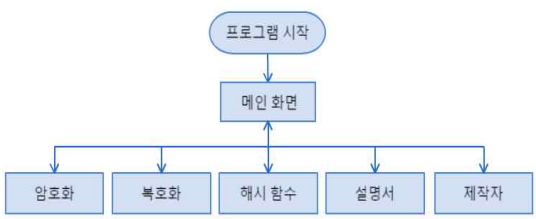
[그림 3.2] 암호화 파일 구조

복호화 기능은 파일 선택, 암호 입력, 경로 설정, 진행, 완료 페이지로 구성된다. 파일 선택 페이지에서는 파일을 선택한다. 암호 입력 페이지에서는 암호를 입력하거나 저장된 암호 파일을 불러올 수 있다. 입력 된 비밀번호와 시그니처에 저장 된 비밀번호의 해시값을 비교하여 암호의 일치와 불일치를 확인한다. 경로 설정 페이지에서는 복호화 하여 생성될 파일의 경로를 선택할 수 있다. 진행 페이지에서는 스레드를 사용하여 시간과 복호화 과정을 동시에 표시하며, 작업이 완료 될 시 복호화 된 파일의 해시값과 시그니처의 가져온 원본 파일의 해시값을 비교한다. 완료 페이지에서는 이전 페이지에서와 해시값의 비교 결과를 표시하며, 생성된 복호화 파일의 위치를 열어주는 항목을 삽입했다.

HASH 기능은 파일 선택, 해시값 확인, 해시값 복사로 구성된다. 파일 선택에서는 암호화된 파일을 불러와 해당 파일의 해시값을 확인하는 기능까지 제공한다. 해시값 복사는 나타난 해시값을 복사하여 메모하거나 다른 프로그램으로 옮길 수 있도록 한다.

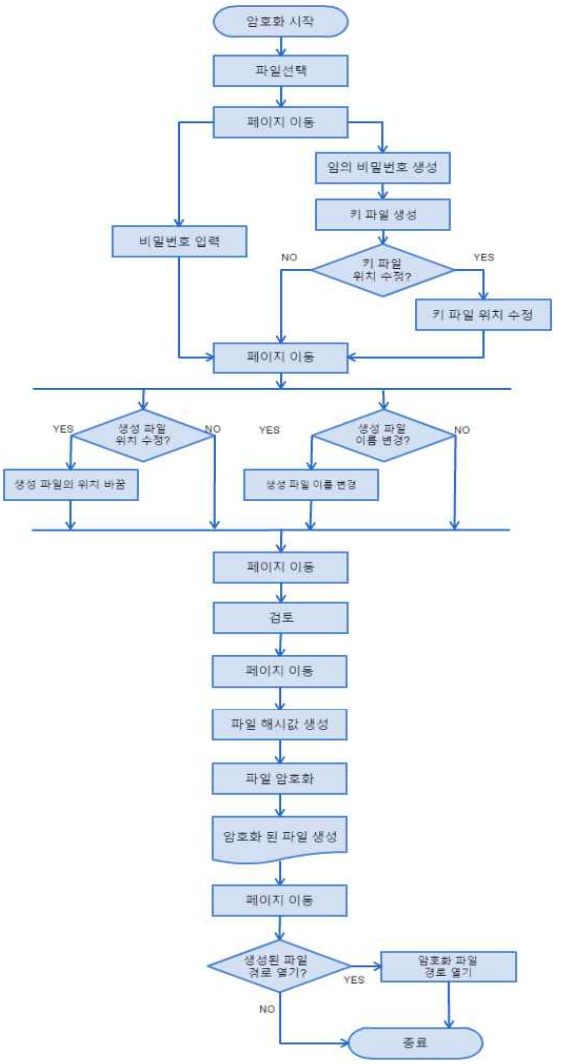
3.2 시스템 구조

시스템 구조 흐름도는 다음과 같다.



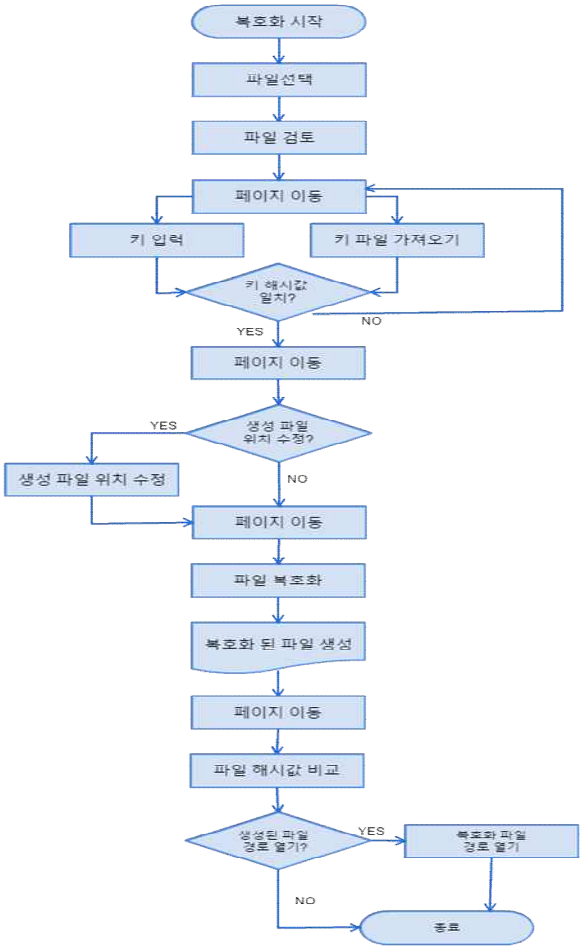
[그림 3.3] 시스템 전체 흐름도

암호화 시작 시 파일 선택하고 페이지를 이동한다. 그 후 비밀번호 임의 생성 또는 사용자 입력을 한다. 임의 생성 시 암호에 대한 pwd 파일을 생성한다. 페이지 이동 후 생성 파일 위치와 이름을 변경할 수 있다. 다음 페이지에서 생성할 파일 내용을 검토한다. 마지막으로 원본 파일 해시 값 생성하고 파일을 암호화 한다.



[그림 3.4] 시스템 암호화 흐름도

복호화 시작 시 파일 선택하고 페이지를 이동한다. 그 후 비밀번호 파일 불러오기 또는 사용자 입력으로 비밀번호를 입력한다. 시그니처에 있는 비밀번호 해시 값으로 비밀번호를 확인한다. 페이지 이동 후 생성 파일 위치 변경할 수 있다. 다음 페이지에서 파일을 복호화 한다. 마지막으로 시그니처에 있는 원본파일의 해시 값과 파일의 해시 값을 비교하여 결과를 출력한다.

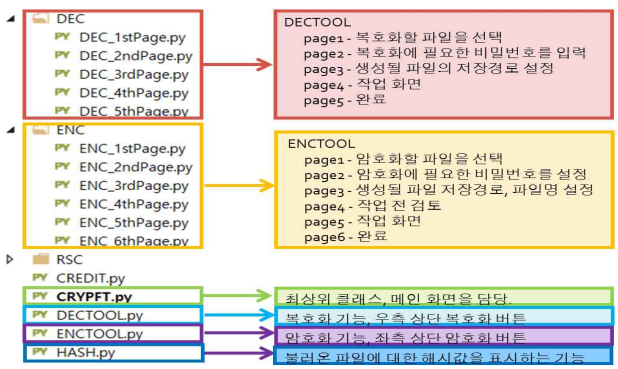


[그림 3.5] 시스템 복호화 흐름도

4. 구현 결과

4.1 시스템 구조 및 개발 환경

처음 시작되는 파일은 CRYPFT.py 파일이다. 그리고 각 페이지 마다 py파일을 작성했고, 페이지에 사용되는 함수를 TOOL이라는 py파일에 작성했다.



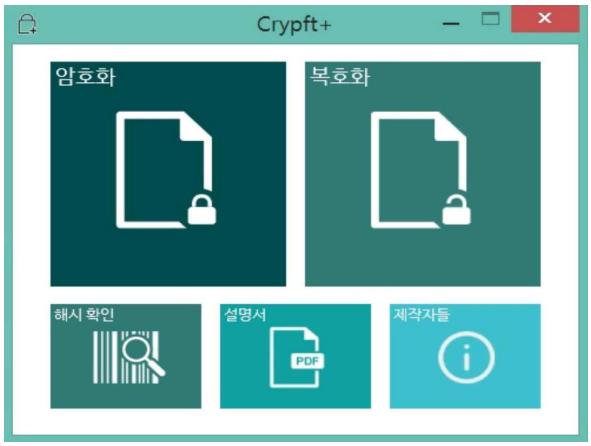
[그림 4.1] py 파일 설명

프로그램을 개발하는데 사용한 프로그램 언어, 코드 작업 툴, 그래픽 작업 툴, 흐름도 작업 툴, 개발한 OS 종류에 대해 표를 작성했다.

[표 4.1] 개발 환경

프로그램 언어	Python 3.4.3 Pyqt5
코드 작업 툴	visual studio 2013/2015
그래픽 작업 툴	Pixlr 웹 에디터
흐름도 작업 툴	cacoo 웹 에디터
OS	Windows 8.1/10 64bit

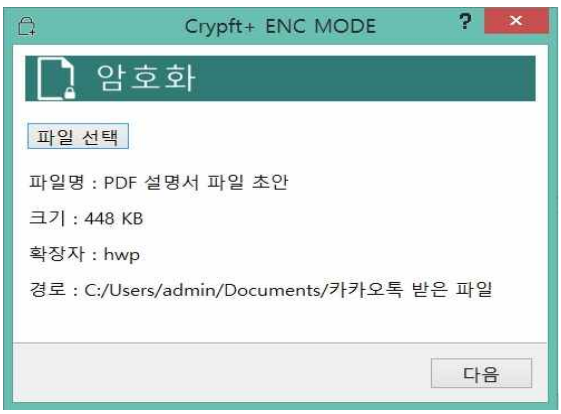
4.2 프로그램 실행



[그림 4.2] 프로그램 메인 화면

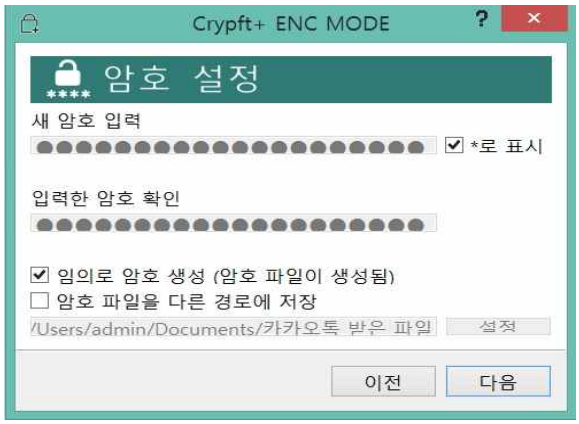
4.2.1 암호화 진행 과정

파일추가 버튼 눌러서 암호화를 진행할 파일을 추가한다.



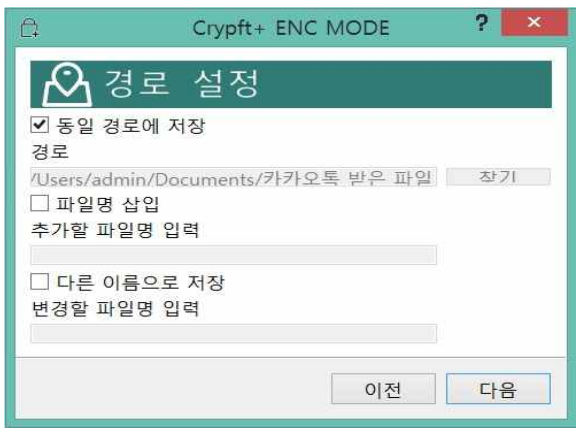
[그림 4.3] 파일 선택 페이지

암호를 입력하거나 임의 암호를 생성한다.



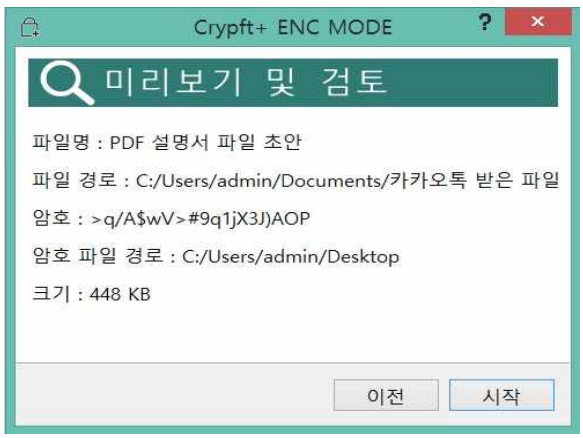
[그림 4.4] 암호 설정 페이지

저장할 경로를 선택 및 다른 이름으로 저장한다.



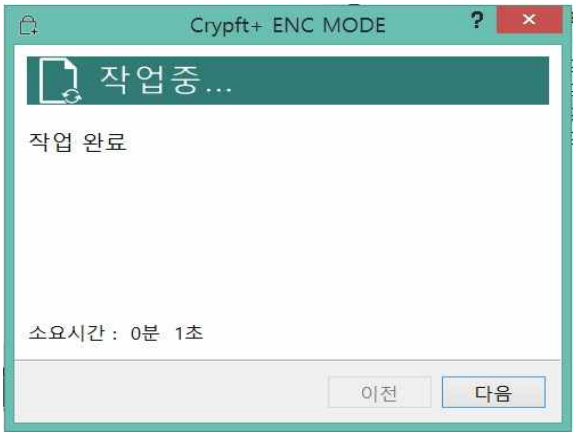
[그림 4.5] 생성 파일 경로 설정 페이지

암호화하기 전 파일에 대한 세부내용을 확인한다.



[그림 4.6] 검토 페이지

암호화 작업을 진행한다.



[그림 4.7] 작업 페이지

완료 페이지 후 파일 위치 열기에 체크로 하고 마치면 해당 파일의 위치가 실행된다. 그리고 암호화 작업 결과를 출력한다.

[그림 4.8] 완료 페이지

암호화 결과로 enc 파일의 암호화된 파일과 임의의 암호 설정으로 인한 pwd 파일이 생성 된다.

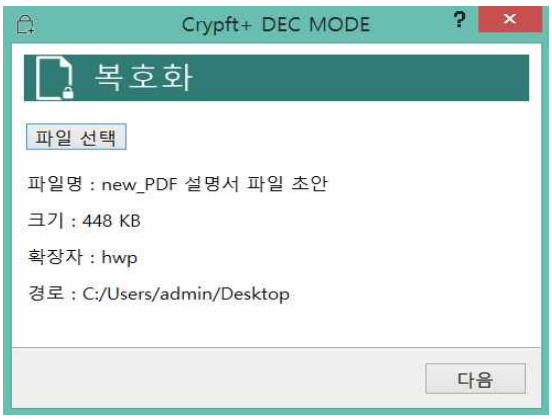
[그림 4.9] 암호화된 파일 생성(enc. pwd 파일)

Crypft+의 약자인 cft와 암호화 대상 파일의 확장자인 hwp 그리고 비밀번호와 원본 파일의 해시값으로 구성되어 있다.

[그림 4.10] 암호화 된 파일 구조(데이터 부분 제외)

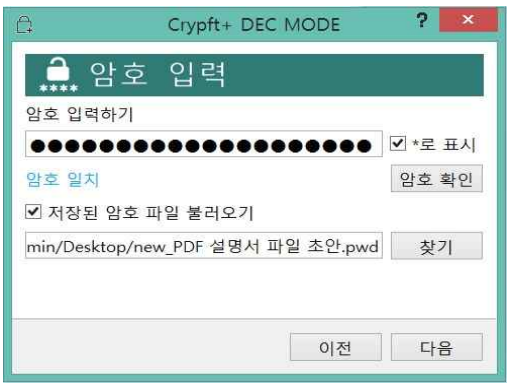
4.2.2 복호화 진행 과정

파일선택을 눌러 복호화를 진행할 파일 추가한다.



[그림 4.11] 파일 선택 페이지

암호를 입력하거나 저장된 암호파일을 불러온다. 암호에 따라 일치 불일치를 표시한다.



[그림 4.12] 암호 입력 페이지(암호 일치)

[그림 4.13] 암호 입력 페이지(암호 불일치)

[그림 4.14]과 [그림 4.15]은 두 개의 해시 값이 있다. 사용자의 입력한 암호의 해시 값, 그리고 시그니처 값에 저장되어 있는 암호 해시 값이다. 두 개의 값을 비교하여 암호의 일치/불일치를 확인한다. [그림 4.12]와 [그림 4.13]으로 확인이 가능하다.

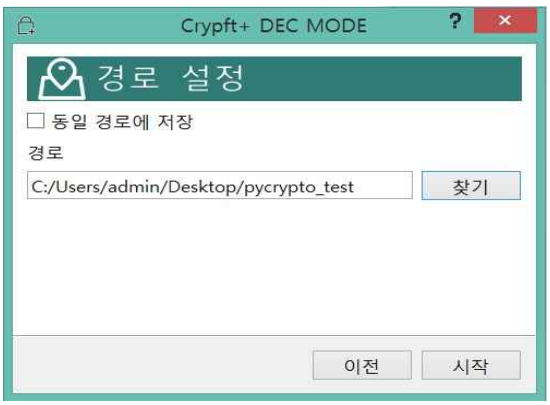
```
f46fcbf0f557eb689356a22e8f5ad8feed34597f007505eb8b4eb091d5b257a6b206391c79f23c58
7af275bbe539219228cb3e952bd3215b9a56f240d5232570
f46fcbf0f557eb689356a22e8f5ad8feed34597f007505eb8b4eb091d5b257a6b206391c79f23c58
7af275bbe539219228cb3e952bd3215b9a56f240d5232570
```

[그림 4.14] 암호 해시 값 비교(일치)

```
66130cf1f956f5ad24957b72186f04de28ff5cd9582c5c7dd25de6dd0f42d3c52a9d954a7cf874c4
6f948dc36a3f886454600c141ff4523540ce49f313aacf34
f46fcbf0f557eb689356a22e8f5ad8feed34597f007505eb8b4eb091d5b257a6b206391c79f23c58
7af275bbe539219228cb3e952bd3215b9a56f240d5232570
```

[그림 4.15] 암호 해시 값 비교(불일치)

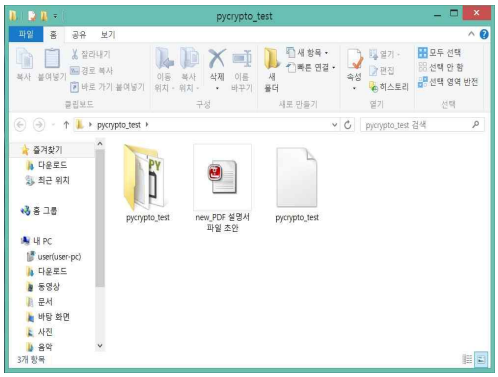
저장할 경로를 선택한다.



[그림 4.16] 경로 설정 페이지

작업 페이지 암호화와 동일하므로 생략한다. 완료 페이지 후 파일 위치 열기에 체크로 하고 마치면 해당 파일의 위치가 실행된다. 그리고 복호화 작업과 파일 손상 여부 결과를 출력한다.

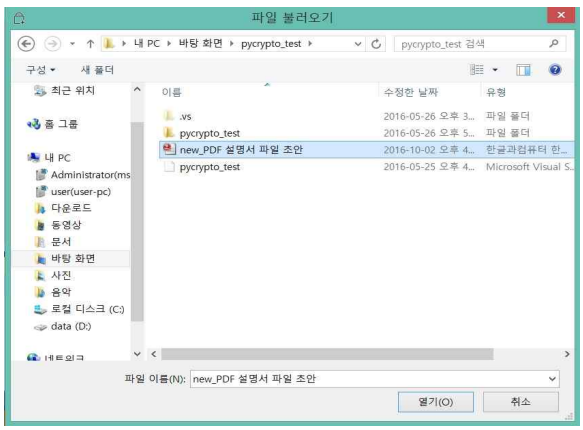
[그림 4.17] 완료 페이지



[그림 4.18] 복호화 완료된 파일

4.2.3 HASH 진행 과정

HASH 기능 버튼을 누르면 파일 불러오기 창이 열린다.



[그림 4.19] 파일 불러오기

파일 선택 후 HASH 결과가 출력되면 해시 복사 버튼을 누르면 복사가 된다.

[그림 4.20] HASH 결과 페이지

5. 결론

본 시스템은 공인된 알고리즘을 이용하여 사용자가 보다 안전하게 사용할 수 있는 환경을 제공하기 위해 개발하였다.

현재 일부 대기업을 제외한 대부분의 중소기업에서는 회사 내 파일공유에 있어 독자적인 프로그램을 쓰지 않거나 보안성이 떨어지는 프로그램을 쓰면서 정보 유출이나 해킹의 위협에 노출되어있는 상황이다. 따라서 누구나 간편하게 사용할 수 있는 Python을 활용한 암복호화 프로그램으로 국내 개인이나 중소기업을 대상으로 손쉬운 인터페이스를 통해 안전한 환경을 제공하도록 하였다. 또한 현재까지 정보 노출의 위협으로부터 안전하게 공인된 알고리즘인 AES를 사용하여 신뢰성을 확보했다.

전체 프로그램 설계는 가볍고 빠르며 간결한 인터페이스를 제공하며 보안성이 뛰어난 알고리즘을 사용하여 정보유출을 막고 유지보수를 용이하게 하기 위한 목적으로 설계하였다. 쉽고 간편한 플랫폼을 이용해 보안성까지 챙길 수 있는 프로그램으로 현재 중소기업들이 노출된 정보 노출의 위협으로부터 보호할 수 있도록 설계하였다.

현재 프로그램은 AES 알고리즘을 사용하고 있지만, 점차 암복호화 알고리즘의 가짓수를 늘려 향후 다양한 암호 알고리즘을 사용할 수 있게 할 예정이다. 또한 기업뿐만 아니라 개인 사용자를 위한 플랫폼을 제작하여 보안성을 널리 퍼뜨려 나갈 예정이다.

REFERENCES

[1] Python https://ko.wikipedia.org/wiki/%ED%8C%8C%EC%9D%B4%EC%8D%AC
[2] PyQt https://opentutorials.org/module/544/4998
[3] Cx-Freezes http://cx-freeze.sourceforge.net/
[4] AES 암호 알고리즘 https://ko.wikipedia.org/wiki/%EA%B3%A0%EA%B8%89_%EC%95%94%ED%98%B8%ED%99%94_%ED%91%9C%EC%A4%80
[5] HASH https://ko.wikipedia.org/wiki/%ED%95%B4%EC%8B%9C_%ED%95%A8%EC%88%98

신 동 호(Dongho Shin)

- 1993년 2월 1일 수원 생
- 2011년 : 숙지고등학교 졸
- 2017년 : 한신대학교 컴퓨터공학부 졸업예정

<관심분야>
정보보안, 데이터 보안, 보안 솔루션 개발

배 우 리(Woori Bae)

- 1992년 10월 24일 서울 생
- 2011년 : 서현고등학교 졸
- 2017년 : 한신대학교 컴퓨터공학부 졸업예정

<관심분야>
서버보안, 서버구축, 사물인터넷 보안

신 형 규(Hyeonggyu Shin)

- 1992년 2월 12일 수원 생
- 2011년 : 유신고등학교 졸
- 2017년 : 한신대학교 컴퓨터공학부 졸업예정

<관심분야>
데이터보안, 정보보안, 데이터베이스 구축

남 승 진(Seungjin Nam)

▪ 1992년 8월 3일 부천 생
▪ 2011년 : 숭실고등학교 졸
▪ 2017년 : 한신대학교 컴퓨터공학부 졸업예정

<관심분야>
정보보안, 사물인터넷, 증강현실

이 형 우(Hyung-Woo Lee) [종신회원]

▪ 1994년 2월 : 고려대학교 컴퓨터학과 (학사)
▪ 1996년 2월 : 고려대학교 컴퓨터학과 (석사)
▪ 1999년 2월 : 고려대학교 컴퓨터학과 (석사)
▪ 2003년 3월 ~ 현재 : 한신대학교 컴퓨터공학부 교수

<관심분야>
사물인터넷, 정보보호, 모바일 보안 및 디지털 포렌식