Visual Studio 환경을 이용한 파일 보안 UI 기능 설계

장승주*
*동의대학교 컴퓨터공학과

Design of the File Security UI Using in the Visual Studio Environments

Seung-Ju Jang*
*Dongeui Univ. Dept. of Computer Engineering

요 약

본 논문에서 제안하는 파일 보안 기능은 암호알고리즘을 이용하여 윈도우 운영체제에서 파일을 안전하게 저장함으로써 허락되지 않은 사용자의 접근을 제한하도록 한다. 암호화하여 저장된 파일은 복호화 알고리즘으로 복호화해서 파일데이터를 읽게 된다. 이러한 기능은 사용자들이 편리하게 사용할 수 있도록 사용자 인터페이스를 설계하여 프로그램으로 구현한다. 보안 기능으로 구현된 파일 암호화 및 복호화 프로그램을 구동시키고 정상적으로 동작하는지의 여부를 실험하게 된다. 또한 복호화시 암호화 할 때의 설정과 설정이 틀 릴 경우 복호화가 되는지의 여부도 실험한다. 이러한 기능을 편리하게 사용할 수 있도록 Visual Studio 환경을 이용하여 UI(User Interface) 기능을 설계한다.

1. 서론

컴퓨터 시스템에서 운영체제를 이용한 데이터 파일의 생성은 급증하고 있는 추세이다. 그러나 운영체제 내의 데이터 파일을 안전하게 관리하는 것은 중요한 문제가 되었다. 윈도우 운영체제 내에서 파일 보안과 관련한 기술을 PDA등 임베디드 시스템 환경에 접목할 경우 중요한 데이터에 안전한 관리를 보장할 수 있다. 윈도우 운영체제 파일 보안 기술은 차세대 컴퓨터 시스템 관련 핵심 기술로써 중요한 의미를 가진다 [1].

보안 운영체제는 기존의 커널에 보안 기능을 통합시킨 보안 커널이 추가로 이식된 운영체제이다. 보안 운영체제의 기능은 사용자에 대한 식별 및 인증, 강제적인 접근 통제, 임의적인 접근 통제, 감사 및 감사 기록, 침입 탐지 등의 기능을 가지고 있다. 파일을 보호하는 기존의 연구 내용으로는 파일의 접근 권한에 대한 엑세스 정보를 관리하여 이루어지는 경우가 있다. USB와 같은 특수 장치내의 파일에 대한 보호는 접근 제어 기법 등을 이용한다. [2, 3].

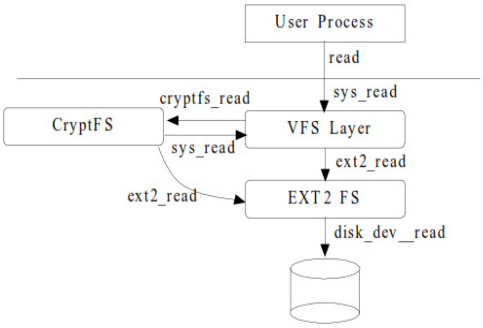
따라서 본 논문에서는 이러한 컴퓨터 시스템 내의 파일에 대한 보안 기능을 제공하여 파일을 안전하게 관리할 수 있도록 파일 보안 UI를 설계한다.

본 논문의 구성은 2장에서는 파일 보안 관련 연구에 대해서 논한다. 3장에서는 Visual Studio를 이용한 파일 보안 프로그램 설계 내용, 4장에서는 실험 및 평가에 대해서 설명하고 5장에서 결론을 논한다.

2. 관련 연구

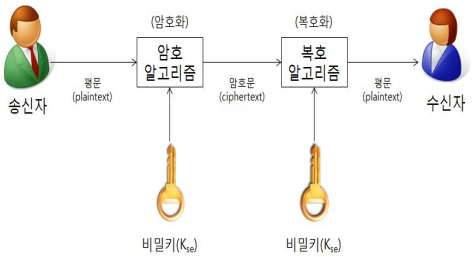
현재 개발되었거나 제안된 보안 파일 시스템들은 크게 두 가지의 형태로 구성되어 있다. 첫 번째는 커널에 포함된 형태이고 다른 형태로는 User- level File System이다. 커널에 포함된 파일 시스템의 경우는 개발이나 디버깅이 어렵다 [4, 5, 6]. 이러한 형태의 파일 시스템을 개발 할 경우에는 low level의 디바이스와 운영체제에 대한 기능을 충분히 이해하고 있어야 하지만 커널에 상주하기 때문에 성능 면에서는 우수하다.

CryptFS는 vnode stacking 기법을 사용하여 모듈화된 계층적인 파일 시스템을 지원하는 메커니즘이다 [6].



(그림 1) CryptFS 파일 시스템 구조

암호화 알고리즘은 암호화 키(*KE*)를 사용하며, 복호화 알고리즘은 복호화 키(*KD*)를 사용한다.

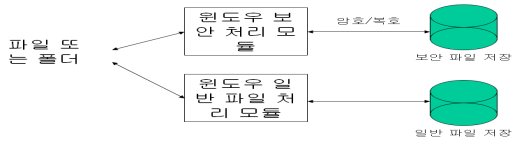


(그림 2) 일반적인 암호, 복호화 과정

비밀키 암호 알고리즘은 SSL(Secure Socket Layer)이나 IPSec(IPSecurity)등 보안 프로토콜 등에서 중요한 역할을 하며 비밀키 암호 알고리즘은 크게 블럭 암호 알고리즘과 스트림 암호 알고리즘으로 나눌 수 있다 [2-3].

3. 운도우 운영체제에서 파일 보안 프로그램 및 UI 설계

본 논문은 중요한 파일에 대해서 지정한 암호화 등을 통해서 보호를 할 수 있도록 한다.



(그림 3) 특정 파일에 대한 보안 수행 과정

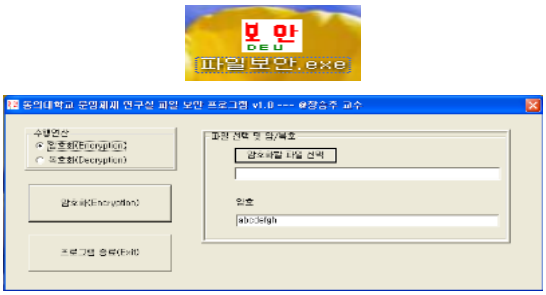
(그림 3)은 윈도우 운영체제에서 특정한 파일에 대한 보안을 수행하는 과정을 보여준다.

파일보안을 위한 기능설계 내용은 다음과 같다

- O 사용자가 지정한 파일에 대한 암호화 기능 설계
- O 사용자가 지정한 파일에 대한 데이터를 버퍼로 읽기 기능 설계
- O 버퍼로 읽어들인 데이터를 암호 알고리즘을 사용하여 암호화 하는 기능 설계
- O 암호화된 버퍼 데이터를 새로운 파일로 저장하는 기능 설계
- O 사용자가 지정한 암호화된 파일에 대한 복호하 기능 설계
- O 복호화를 하기 위하여 암호화된 파일로부터 버퍼로 데이터를 읽어들이는 기능 설계
- O 버퍼로 읽어들인 암호화 데이터를 복호 알고리즘을 사용하여 복호하 하는 기능 설계
- O 버퍼 내의 복호화된 데이터를 파일에 저장하는 기능 설계

3.1 파일 암호화 대상 파일 선택 기능 및 UI 설계

본 논문에서 사용자가 편리하게 사용할 수 있도록 설계된 화면은 다음 (그림 4)와 같다.

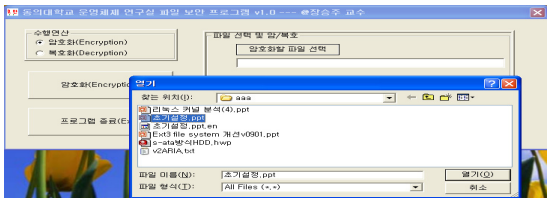


(그림 4) 보안 프로그램 실행 UI 화면

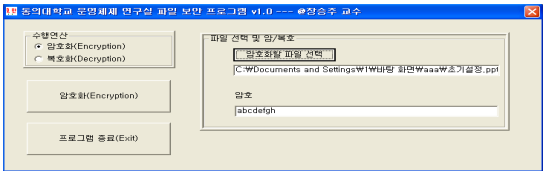
파일 보안 프로그램 동작 화면에서 수행 연산 부분은 파일에 대한 암호, 복호화 여부를 결정하는 기능이다.

3.2 사용자가 지정한 파일에 대한 암호화 기능 및 UI 설계

사용자가 지정한 파일에 대한 암호화기능이다.



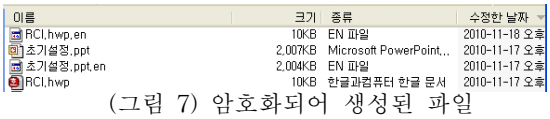
(그림 5) 암호화할 파일 선택 UI 화면



(그림 6) 암호화할 파일 경로 설정 UI

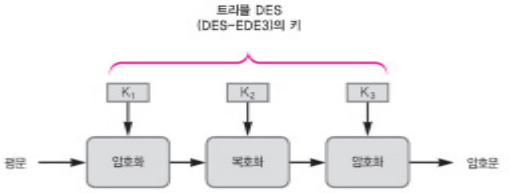
(그림 6)과 같이 설정 끝나면 암호화

(Encryption) 버튼을 누르게 되면 해당 파일이 암호화 되게 된다. (그림 5), (그림 6)은 암호화할 파일을 선택하는 과정이다.

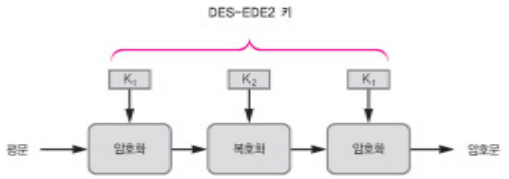


(그림 7) 암호화되어 생성된 파일

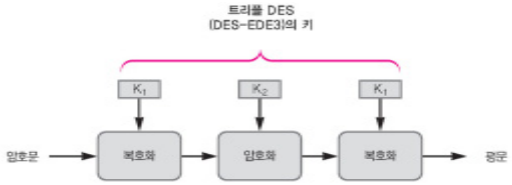
새롭게 암호화되어 생성된 파일은 확장자가 “en”이 추가되게 된다. 일반 사용자 파일을 암호화 및 복호화할 경우에 사용하는 암호 알고리즘은 triple-DES를 사용한다.



(a) triple-DES 알고리즘의 암호화 과정



(b) DES-EDE2 과정



(c) triple-DES 알고리즘 복호화 과정

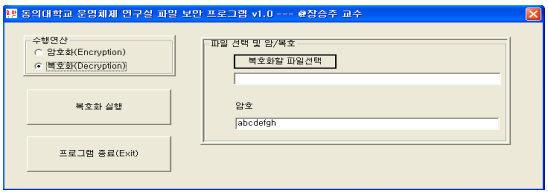
(그림 8) triple-DES 암호/복호화 과정

위 (그림 8)은 triple-DES 알고리즘의 동작 과정을 나타낸다. 64비트 평문은 초기 자리바꿈(IP, Initial Permutation) 과정을 거친 후에 두 개의 32비트 블록으로 나누어진다.

임의의 입력을 초기 자리바꿈을 한 다음에 다시 최종 자리바꿈을 하면 그 결과는 원 입력과 같아진다.

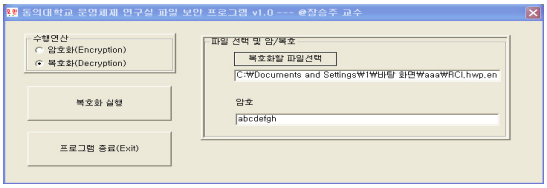
3.3 사용자가 지정한 암호화된 파일에 대한 복호화 기능 및 UI설계

복화화 과정은 앞에서 설명한 암호화 과정의 역순으로 일어나게 된다.



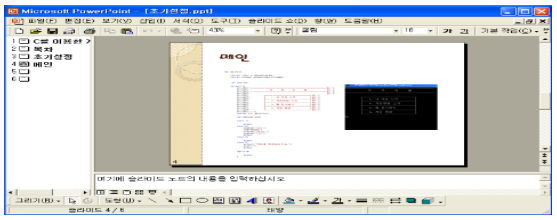
(그림 9) 암호화된 파일을 복호화된 UI 화면

위 (그림 9)와 같이 암화화된 파일에 대해서 복호화가 실행되게 된다. 복호화할 파일 선택 아래에 (그림 10)과 같이 경로가 나타난다.



(그림 10) 복호화할 파일 경로 설정 UI 화면

(그림 10)과 같이 복호화할 파일이 지정되고 실행 버튼을 누르면 암호화된 파일이 복호화되게 된다.

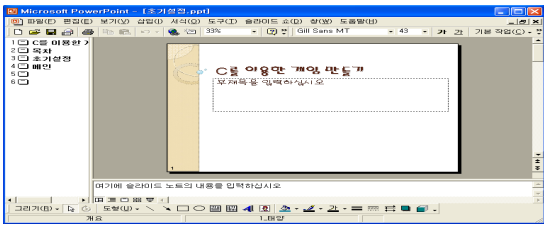


(그림 11) 복호화한 후 파일 열기 UI 화면

(그림 11)은 암호화한 파일을 복호화를 실행하고 난후의 파일 실행 결과화면이다. 이 화면에서 보면 암호화 하기 전의 내용이 정상적으로 보임을 확인할 수 있다.

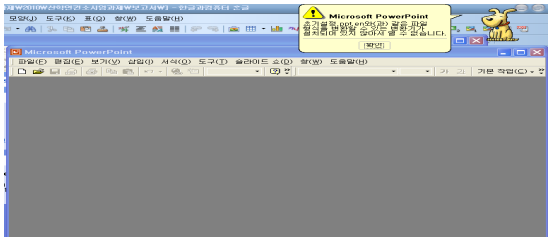
4. 실험 및 평가

본 논문에서 제안한 내용을 구현한 윈도우 운영체제에서 보안 프로그램의 기능은 크게 두가지로 나누어진다. 하나는 일반 파일을 허가되지 않은 사용자가 접근하지 못하도록 차단하는 암호화 기능이다. 다른 하나는 허가된 사용자가 암호화된 파일을 복호화하여 정상적인 파일로 볼 수 있도록 해 주는 기능이다.



(그림 12) ppt 파일 읽기 화면

(그림 12)는 일반적으로 많이 사용하는 ppt 파일을 “열기“하여 읽은 화면을 보여준다.



(그림 13) 암호화된 ppt 파일을 “열기”했을 경우 오류 메시지가 발생되는 화면

(그림 13)은 암호화된 ppt 파일을 “열기”할 경우 오류 메시지가 발생되는 화면이다. 오류가 발생되는 이유는 파일을 암호화했기 때문에 정상적인 데이터를 “파워포인터(ppt)” 프로그램에서 읽을 수 없기 때문이다. ppt 파일을 암호화하게 되면 보안 프로그램에서 파일의 확장자로 “.en”을 붙이도록 설계되어 있다. 파일의 확장자 명 자체가 잘못인식기 되기 때문이다.

암호화된 파일에 대해서 본 논문에서 설계한 프로그램을 사용하여 사용자가 해당 파일을 복호화 한다. 복호화된 파일을 열어서 수행한 화면은 위에서 보인 (그림 13)과 같이 수행된다. 이와 같이 일반 파일을 개발한 암호화 기능을 이용하여 암호화하게 되면 허가된 사용자 외에는 파일을 정상적으로 읽을 수 없음을 확인할 수 있다.

위 실험 결과에서 보듯이 본 논문에서 설계 및 개발한 윈도우 운영체제에서 파일 보안 프로그램을 이용하여 보안을 수행한 후 허가된 사용자 외에는 접근은 가능하지만 파일 데이터를 읽기가 불가능함을 알 수 있다. 허가된 사용자의 경우는 암호화된 파일을 본 논문에서 개발한 프로그램을 이용하여 복호화함으로써 정상적으로 읽기가 가능함을 확인할 수 있었다.

5. 결론

본 논문에서 제안한 내용을 실제 구현하여 실험을 수행하였다. 실험은 윈도우 운영체제가 탑재된 시스템 환경에서 본 논문의 구현 프로그램을 이용하여 이루어졌다. 본논문에서 제안하는 기능을 편리하게 사용할 수 있도록 하는 UI 기능도 설계했다. 본 논문에서 제안하는 기능으로 윈도우 운영체제에서 보안 프로그램의 기능은 크게 두 가지 이다. 하나는 일반 파일을 허가되지 않은 사용자가 접근하지 못하도록 차단하는 암호화 기능이다. 다른 하나는 허가된 사용자가 암호화된 파일을 복호화 하여 정상적인 파일로 볼 수 있도록 해 주는 기능이다. 실험 결과 본 논문에서 구현한 파일 보안 기능이 정상적으로 동작함을 확인할 수 있었다. 또한, 기존의 연구 결과는 사용상의 불편한 등이 있지만, 간편한 사용자 인터페이스와 검증된 암호 및 복호 알고리즘의 사용으로 안정성을 보장하고 있다.

참고문헌

[1] 국가정보원, 2009 국가정보보호백서, 제2편 제6장 개인정보보호 활동, 2009년 4월

[2] 김용관(Yongkwan Kim) 위영철(Youngcheul Wee), "휴대장치를 위한 고속복원의 프로그램 코드 압축기법", 정보과학회논문지 : 소프트웨어 및 응용, Vol.37 No.11, 2010.

[3] 전창규(Chang Kyu Jeon) 류경식(Kyeung Seek Lew) 김용득(Yong Deak Kim), “임베디드 시스템에서 실행 가능 압축 기법을 이용한 프로그램 초기 실행 속도 향상”, 電子工學會論文誌-CI, Vol.49 No.1, 2012.

[4] 이성헌(Seong-Heon Lee) 장승주(Seung-Ju Jang), “데이터 압축을 통한 효율적인 저장 공간 사용을 보장하는 저널링 파일 시스템 설계”, 한국정보과학회 학술발표논문집, Vol.38 No.2A, 2011.

[5] 조범석(Beom-Seok Joh) 김영로(Young-Ro Kim), “가변적 준무손실 압축 알고리즘 = Variable Near-Lossless Compression Algorithm”, 대한전자공학회 학술대회 논문집, Vol.2010 No.6, 2010.

[6] 조미남(Mi-Nam Cho) 지유강(Yoo-Kang Ji), "임베디드시스템을 위한 혼용텍스트 파일의 개선된 LZW 압축 알고리즘 구현", 한국콘텐츠학회논문지, Vol.10 No.12, 2010