윈도우 운영체제에서 암호화 알고리즘을 이용한 파일 보안 기능 설계

장승주*

Design of the File Security Function Using Encryption Algorithm in the Windows Operating System

Seung-Ju Jang*

요 약

본 논문에서 제안하는 파일 보안 기능은 암호 알고리즘을 이용하여 윈도우 운영체제에서 파일을 안전하게 저장함으로써 허락되지 않은 사용자의 접근을 제한하도록 한다. 암호화하여 저장된 파일은 복호화 알고리즘으로 복호화해서 파일 데이터를 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 복호화 시 암호화 할 때의 설정과 설정이 틀릴 경우 복호화가 되는지의 여부도 실험한다. 이 프로그램의 개발을 통해서 윈도우 서버 및 개인용 컴퓨터 내의 중요한 파일에 대한 보안을 강화시킬 수 있다.

ABSTRACT

The file security function, which this paper suggests, restricts the access of an unauthorized users by using password algorithm and saving file. Saved files that are encrypted are read by decrypting them with decryption algorithm. These features are user interface to design the program for user friendly. The security function implements both file encryption and decryption programs and tests whether the experiment works or not. In addition, when a decryption is progressed and the settings of between decryption and encryption are different each other, the security function also checks the possibility of decryption. We can enhance the security on important files stored in Windows servers or personal computers by developing this program.

키워드

파일 보안, 모듈, 파일 암호 알고리즘, 윈도우 운영체제, 복호화 모듈

key word

File Security, Module, File Encryption Algorithm, Windows O.S, File Decryption Module

* 정회원 : 동의대학교 컴퓨터공학과 교수(교신저자, sjjang@deu.ac.kr)

접수일자 : 2012. 10. 19
심사완료일자 : 2012. 12. 05

Open Access http://dx.doi.org/10.6109/jkiice.2013.17.3.612

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

컴퓨터 시스템에서 운영체제를 이용한 데이터 파일의 생성은 급증하고 있는 추세이다. 그러나 운영체제 내의 데이터 파일을 안전하게 관리하는 것은 중요한 문제가 되었다. 특히, 스마트 폰 시스템 환경의 급속한 보급은 이러한 기능의 필요성이 증대되고 있다. 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]. 컴퓨터 시스템 보안은 정보화 사회가 되면서 중요한 이슈가 되고 있다. 최근에는 보안 운영체제에 대한 연구가 활발히 진행되고 있다. 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. 이러한 보안 운영체제에서 파일의 보안은 더욱 중요하다. 파일을 보호하는 기존의 연구 내용으로는 파일의 접근 권한에 대한 엑세스 정보를 관리하여 이루어지는 경우가 있다. 또한, USB 장치와 같은 특수 장치 내에 저장된 파일에 대한 보안을 위한 연구가 진행되고 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 한다. 이러한 개발을 통해서 보다 안정된 운영체제 파일 보안 기능의 제공으로 파일의 관리가 보다 안전하게 이루어질 수 있도록 하고자 한다. 본 논문에서 제안하는 파일 보안 기능은 암호 알고리즘을 이용하여 윈도우 운영체제에서 파일을 안전하게 저장함으로써 허락되지 않은 사용자의 접근을 제한하도록 한다. 본 논문은 기존의 논문과 차별적으로 편리한 사용자 인터페이스와 간결한 동작 구조, 안정성이 보장된 암호화 및 복호화 알고리즘을 이용하여 구현한다. 암호화하여 저장된 파일은 복호화 알고리즘으로 복호화해서 파일 데이터를 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 실험한다. 본 논문의 구성은 2장에서는 파일 보안 관련 연구에 대해서 논한다. 3장에서는 파일 보안 프로그램 설계 내용, 4장에서는 실험 및 평가에 대해서 설명하고 마지막으로 결론을 논한다.

Ⅱ. 관련 연구

보안 파일시스템은 1993년 AT&T Bell 연구소의 Blaze에 의해서 개발된 CFS (Cryptographic File System)에 의해서 제안되었다 [1,3]. 기존에 여러 가지의 보안 도구들이 존재했지만 운영체제 레벨에서 암호화 기능을 제공하는 시도는 근래에 들어서 이루어졌다. 이후 CFS의 기능을 보완한 TCFS(Transparent Cryptographic File System)등의 보안 파일시스템들이 개발되었다 [3].

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. 반면, User-level에 구현된 파일 시스템은 구현이 용이한 반면 성능 면에서는 뒤떨어진다.

CryptFS는 vnode stacking 기법을 사용하여 모듈화된 계층적인 파일 시스템을 지원하는 메커니즘이다 [7-10]. 디지털 동영상 서비스의 안전성과 보안성을 위해서 암호기술이 필요하다. 암호 기술은 특정 내용을 정해진 사람만이 알 수 있도록 하기 위하여 개발되기 시작했다.

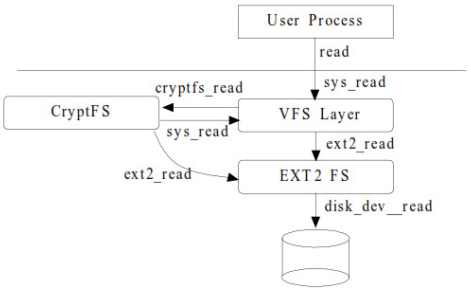


그림 1. CryptFS 파일 시스템 구조
Fig. 1 CryptFS File System Architecture

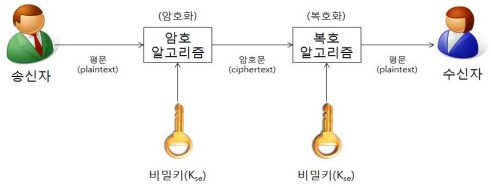


그림 2. 일반적인 암호, 복호화 과정
Fig. 2 General Encryption, Decryption Procedure

암호 알고리즘은 암호화 알고리즘(E)과 복호화 알고리즘(D)으로 구성된다. 암호화 알고리즘은 암호화 키(KE)를 사용하며, 복호화 알고리즘은 복호화 키(KD)를 사용한다. 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 [2-5].

Ⅲ. 운도우 운영체제에서 파일 보안 프로그램 설계

본 논문은 윈도우 운영체제의 중요한 파일에 대한 보안 모듈 개발을 목표로 한다. 기존 윈도우 운영체제 내의 파일을 보호하고자 한다. 기존의 윈도우 운영체제는 중요한 데이터 파일에 대한 보호 기능이 없다. 본 논문은 이러한 중요한 파일에 대해서 사용자가 지정을 하면 암호화 등을 통해서 보호를 할 수 있도록 한다. 사용자가 필요할 경우 이 기능을 하는 모듈을 이용하여 사용할 수 있도록 개발한다. 또한 사용자의 편리성을 위해서 쉽게 사용이 가능하도록 한다 [5, 11]. 윈도우 운영체제 내에서 중요한 데이터를 보관하는 파일에 대한 보안을 하는 것은 아주 중요한 일이다. 본 논문은 윈도우 운영체제에서 중요한 파일에 대한 보안 기술을 개발한다. 개발되는 기술은 다음 그림 3과 같이 동작된다.

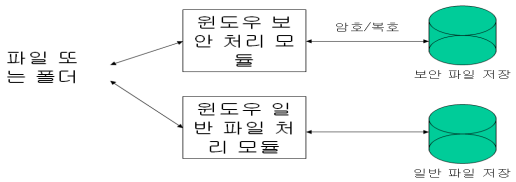


그림 3. 특정 파일에 대한 보안 수행 과정
Fig. 3 Security Procedure for Specific File

그림 3은 윈도우 운영체제에서 특정한 파일에 대한 보안을 수행하는 과정을 보여준다. 일반 파일을 지정하여 보안 파일로 변환하는 과정이다. 사용자가 보호하고자하는 파일을 지정하면 보안 프로그램이 동작되어서 해당 파일을 암호화 등을 수행하여 보호하게 된다.

본 논문에서 파일 보안을 위한 기능 설계 내용은 다음과 같다.

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

본 논문에서는 위와 같은 기능을 중심으로 윈도우 운영체제 환경에서 파일 보안 기능을 설계하였다.

3.1. 파일 암호화 대상 파일 선택 기능 설계

본 논문에서 설계한 윈도우 운영체제에서 파일 보안 기능을 사용자가 편리하게 사용할 수 있도록 사용자 인터페이스를 설계한다. 사용자가 편리하게 사용할 수 있도록 설계된 화면은 다음 그림 4와 같다. 본 프로그램을 실행 하고자 할 경우에 아래의 모양과 같은 아이콘을 두 번 누리게 되면 실행이 되게 된다. 파일 보안 프로그램을 실행하게 되면 다음과 같은 화면이 동작하게 된다.

파일 보안 프로그램 동작 화면에서 수행 연산 부분은 파일에 대한 암호, 복호화 여부를 결정하는 기능이다. 먼저 사용자가 원하는 연산이 암호화일 경우는 암호화 버튼을 누른다. 그리고, 파일 선택 및 암/복호에서 암호화할 파일 선택 부분에서 암호화하고자 하는 파일을 선택한다. 복호화 기능은 암호화된 파일을 정상적으로 되돌려놓고자 할 경우에 사용하는 기능이다. 세부적인 기능

은 아래에 자세히 설명한다. 본 논문에서 제안하는 프로그램을 실행할 경우에 내부적인 동작 과정은 그림 4와 같다.

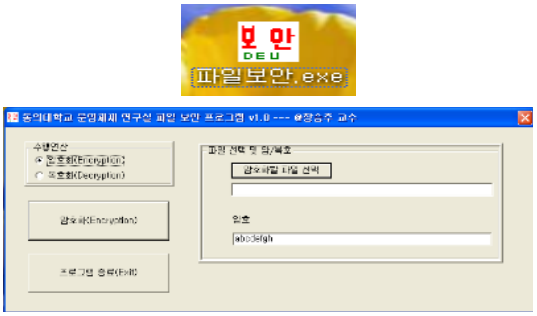


그림 4. 보안 프로그램 실행 화면
Fig. 4 Execution Screen of the Security Program

3.2. 사용자가 지정한 파일에 대한 암호화 기능 설계

사용자가 지정한 파일에 대한 암호화 기능은 수행연산에서 암호화 버튼을 누르고 파일 선택 및 암/복호에서 암호화할 파일을 선택하게 된다. 그러면 아래와 같이 파일 선택을 할 수 있는 창이 뜨게 된다. 그러면 자신이 암호화하고자 하는 파일을 선택하면 된다. 아래 화면은 RCI.hwp 파일을 선택하는 화면이다.

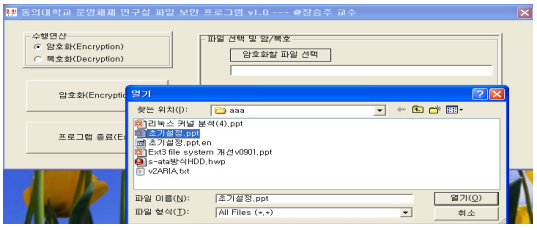


그림 5. 암호화할 파일 선택 화면
Fig. 5 File Selection for Encryption File

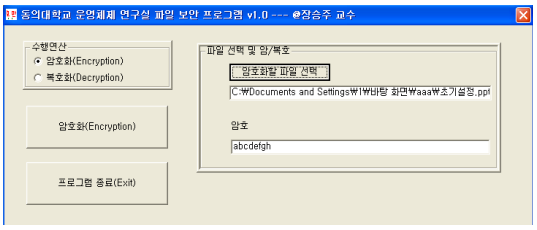
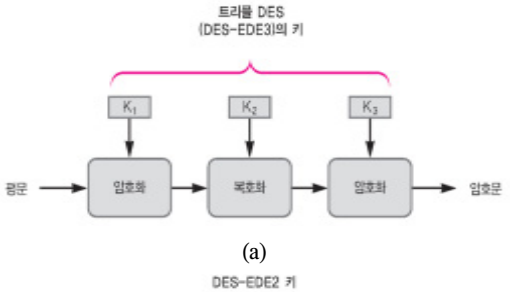


그림 6. 암호화할 파일 경로 설정
Fig. 6 Configuration of the Encryption File Path

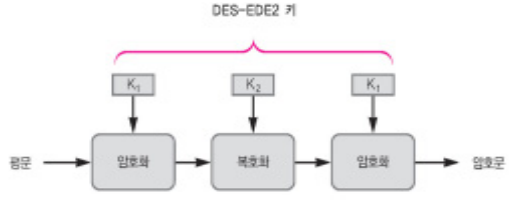
그림 6과 같이 설정이 끝나면 좌측 중간에 있는 암호화(Encryption) 버튼을 누르게 되면 해당 파일이 암호화되게 된다. 지정된 파일이 암호화되게 되면 지정된 폴더 내에 그림 7과 같이 새롭게 암호화되어 생성된 파일이 만들어지게 된다. 그림 5, 그림 6은 암호화할 파일을 선택하는 과정이다.

그림 7. 암호화되어 생성된 파일
Fig. 7 Encrypted File

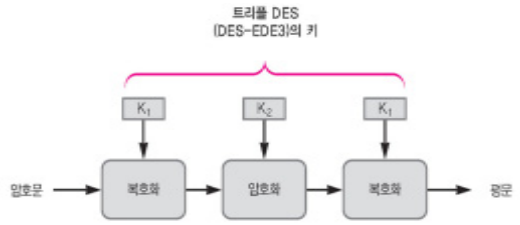
새롭게 암호화되어 생성된 파일은 확장자가 “en”이 추가되게 된다. 일반 사용자 파일을 암호화 및 복호화할 경우에 사용하는 암호 알고리즘은 triple-DES 를 사용한다. triple-DES 알고리즘은 DES보다 강력하도록 DES를 3단 겹치게 한 암호 알고리즘이다. 금융권에서는 전자지불시스템 등에서 DES-EDE2는 아직 상당히 많이 사용되고 있다. 상당기간 암호표준으로 활용될 것으로 여겨진다. triple-DES 알고리즘의 처리 속도는 빠르지 않고 현재 암호 표준으로 지정된 AES 알고리즘보다 속도가 6배 정도가 더 빠르다. AES 알고리즘 보다 완벽한 안전성을 가지고 있으며, 블록의 크기도 더 크고, 키의 길이도 더 길며, 2006년 현재까지도 공개적으로 알려진 암호학적인 공격이 없었다. 다음은 triple-DES 알고리즘의 암호화 및 복호화 과정을 나타낸다.



(a)



(b)



(c)

그림 8. triple-DES 암호/복호화 과정. (a) triple-DES 알고리즘의 암호화 과정 (b) DES-EDE2 과정 (c) triple-DES 알고리즘 복호화 과정
Fig. 8 Encryption/Decryption Procedure Using triple-DES (a) Encryption Procedure of the triple-DES Algorithm (b) DES-EDE2 Procedure (c) Decryption Procedure of the triple-DES Algorithm

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

이들은 16라운드를 거치게 되며, 두 결과 블록은 다시 합쳐 최종 자리바꿈(FP, Final Permutation) 과정을 거쳐 암호문으로 변환된다. 여기서 초기 자리바꿈과 최종 자리바꿈은 서로 역 관계가 성립한다. 즉, 임의의 입력을 초기 자리바꿈을 한 다음에 다시 최종 자리바꿈을 하면 그 결과는 원 입력과 같아진다.

3.3. 사용자가 지정한 암호화된 파일에 대한 복호화 기능 설계

복호화를 수행하기 위해서는 수행연산에서 복호화 버튼을 선택한다. 복화화 과정은 앞에서 설명한 암호화 과정의 역순으로 일어나게 된다. 사용자가 암호화한 파일에 대해서 정상적으로 파일 열기를 할 경우에 먼저 암호화된 파일을 복호화해야 한다. 복호화 기능을 실행하게 되면 그림 9와 같은 사용자 화면이 나타난다.

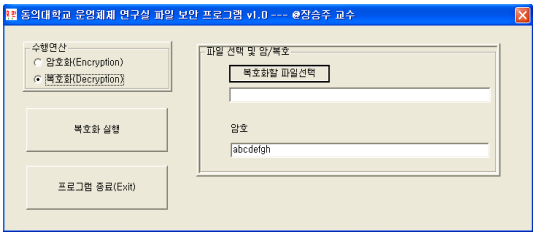


그림 9. 암호화된 파일을 복호화된 화면
Fig. 9 Decrypted Screen for the Encrypted File

위 그림 9와 같이 암화화된 파일에 대해서 “수행연산”으로 “복호화”를 선택한후 “복호화할 파일선택”을 누르면 복호화할 파일을 선택할 수 있다. “복호화할 파일선택”으로 해당 파일을 선택하고 나서 암호를 입력한다. 이 과정의 수행을 마치고 나면 “복호화 실행” 버튼을 누르게 되면 복호화가 실행되게 된다. 복호화할 파일 선택 아래에 그림 10과 같이 경로가 나타난다.

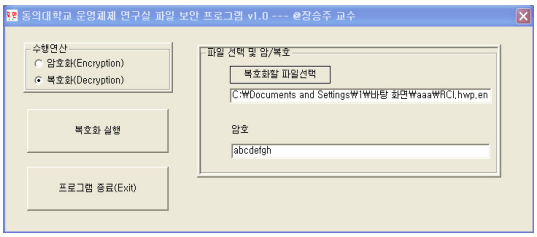


그림 10. 복호화할 파일 경로 설정 화면
Fig. 10 Decrypted File Path Configure Screen

그림 10과 같이 복호화할 파일이 지정되고 나면 왼쪽 중간의 복호화 실행 버튼을 누르면 암호화된 파일이 복호화되게 된다. 암호화된 파일을 복호화 한후에 해당 파일 열기를 통해서 정상적으로 파일이 복호화 되었는지를 확인하게 된다. 아래 그림 10은 정상적으로 파일이 열린 화면을 보여준다 [6, 7].

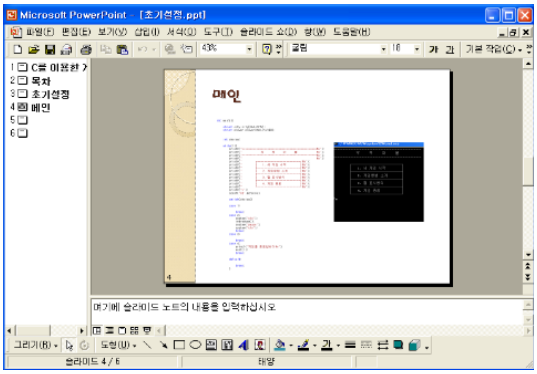


그림 11. 복호화한 후 파일 열기 화면
Fig. 11 File Open Screen after Decrypting

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

Ⅳ. 실험 및 평가

본 논문에서 구현한 결과를 실제 시스템을 이용하여 실험을 수행하였다. 실험을 위해서 사용된 시스템 환경은 다음과 같다.

- O 인텔 팬티엄4 CPU 3.00 GHZ
- O 1.37GB 메인 메모리
- O Microsoft Windows XP 운영체제
- O Visual Studio 6.0 compiler 환경

본 논문에서 제안한 내용을 구현한 윈도우 운영체제에서 보안 프로그램의 기능은 크게 두가지 로 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.

먼저 일반 파일을 허가된 사용자만 볼 수 있도록 암호화하는 기능에 대한 실험 결과를 보인다. 정상적인 파일을 “열기“ 했을 때의 화면은 아래 그림 12와 같다.

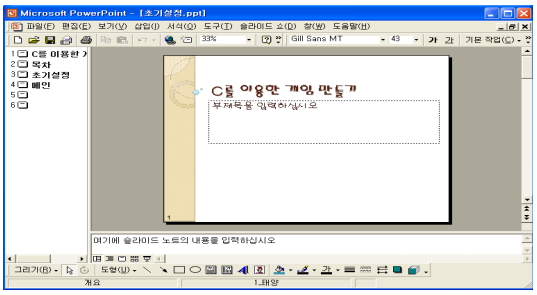


그림 12. ppt 파일 읽기 화면
Fig. 12 PPT File Reading Screen

그림 12는 일반적으로 많이 사용하는 ppt 파일을 “열기“하여 읽은 화면을 보여준다. 일반 파일에 대해서 사용자가 암호화 과정을 수행한후 암호화된 파일을 ”열기“ 했을 경우의 수행 화면은 다음 그림 13과 같다.

그림 13은 암호화된 ppt 파일을 “열기”할 경우 오류 메시지가 발생되는 화면이다. 오류가 발생되는 이유는 파일을 암호하했기 때문에 정상적인 데이터를 “파워포인터(ppt)” 프로그램에서 읽을 수 없기 때문이다. ppt 파일을 암호화하게 되면 보안 프로그램에서 파일의 확장자로 “.en”을 붙이도록 설계되어 있다. 파워포인터 읽기 프로그램은 확장자가 ppt인 파일에 대해서 열기가 가능하기 때문에 위와 같은 오류 메시지가 출력되게 된다. 즉, 파일의 확장자 명 자체가 잘못인식기 되기 때문이다. 그리고, 확장자를 ‘ppt’로 하더라도 파일이 가지고 있는 구조 자체가 암호화로 인하여 변경되었기 때문에 정상적으로 데이터를 읽을 수 없다.

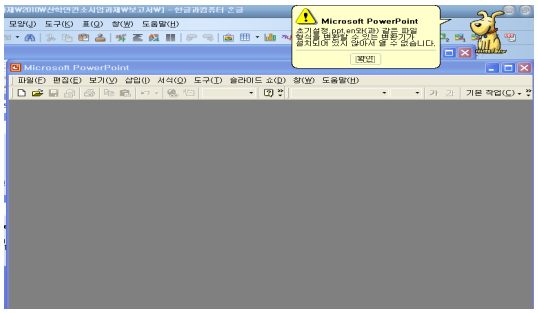


그림 13. 암호화된 ppt 파일을 “열기”했을 경우 오류 메시지가 발생되는 화면
Fig. 13 Error Message for Encryption File Open

암호화된 파일에 대해서 본 논문에서 설계한 프로그램을 사용하여 사용자가 해당 파일을 복호화 한다. 복호화된 파일을 열어서 수행한 화면은 위에서 보인 그림 13과 같이 수행된다. 이와 같이 일반 파일을 본 논문에서 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.

Ⅴ. 결 론

본 논문에서는 윈도우 운영체제에서 사용 가능한 파일 보안 기능을 설계 및 구현하였다. 구현된 보안 기능은 사용자가 윈도우 운영체제 내의 파일에 대해서 허가된 사용자 외에 접근을 차단시키고자 할 경우에 사용이 가능하다. 본 논문에서 개발한 보안 기능의 상세 내용은 윈

도우 운영체제 내의 파일에 대한 보안 모듈 설계, 윈도우 운영체제 내의 파일에 대한 보안 모듈 개발, 기존 윈도우 운영체제 내의 파일과의 연동 모듈 시험, 개발 모듈 통합 및 통합 시험 등의 기능으로 구성되어 있다.

본 논문에서 제안한 내용을 실제 구현하여 실험을 수행하였다. 실험은 윈도우 운영체제가 탑재된 시스템 환경에서 본 논문의 구현 프로그램을 이용하여 이루어졌다. 본 논문에서 구현한 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 해주는 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] Uppuluri, P., Pittges, J., A Comprehensive Undergraduate Application Security Project, Information Technology: New Generations (ITNG), 2012 Ninth International Conference on, pp. 600 - 607, 2012.

[3] Huanan Liu; Shiqing Wang, Research for security strategy of cloud service based on system survivability, Control and System Graduate Research Colloquium (ICSGRC), pp. 1 - 4, 2012.

[4] Steward, C.; Wahsheh, L.A.; Ahmad, A.; Graham, J.M.; Hinds, C.V.; Williams, A.T.; DeLoatch, S.J., "Software Security: The Dangerous Afterthought", Information Technology: New Generations (ITNG), 2012 Ninth International Conference on, pp. 815 - 818, 2012.

[5] Yinghua Wu, Jianping Wu, Ke Xu, Mingwei Xu, "The Design And Implementation of Router Security Subsystem Based on IPSec", IEEE Computer Society, Vol 1. pp.160-165, 8.2002.

[6] Dieter Gollmann, Computer Security, WILEY, Vol, 2. pp.234-251, 2006.

[7] 한국정보보호진흥원, 중소기업 정보보호예상 피해유형과 대응사례, 2006.11

[8] 한국정보보호진흥원, 2008 정보시스템 해킹·바이러스 현황 및 대응, 2008.12

[9] SANS Institute, 20가지 가장 치명적인 보안 위협, 2007.

[10] Apspect Security, Starting Out With Application Security, http://www.aspectsecurity.com/owasp.htm, 2007.

[11] 이성헌, 장승주, "윈도우 운영체제의 파일 보안 모듈 개발", 한국해양정보통신학회 2011년도 춘계학술대회, 2011. 5.

저자소개

장승주(Jang, Seung Ju)

1985년 부산대학교 계산통계학(전산학) 학사
1991년 부산대학교 계산통계학(전산학) 석사
1996년 부산대학교 컴퓨터공학 박사
1987년~1996년 한국전자통신연구원(ETRI) 시스템 SW연구실
1993년~1996년 부산대학교 시간강사
2000년~2002년 Univ. of Missouri at Kansas City, visiting professor
1996년 ~ 현재 동의대학교 컴퓨터공학과 교수
※관심분야 : 운영체제, 임베디드 운영체제, 분산 시스템, 시스템 보안, 스마트 폰 시스템 운영체제