2차 인증방식을 이용한 USB보안 솔루션 (USS Solution)

고상현*, 한석진*, 최윤수*, 배종수*, 이현*
*선문대학교 컴퓨터공학부
e-mail: sanghyun.ko92@gmail.com, 0626na@gmail.com, mahyun91@sunmoon.ac.kr

USB security solution using 2 fator authentication

Sang-Hyun Ko*, Seok-Jin Han*, Yoon-SU Choi*, Jong-Su Bae* and Hyun Lee*
*Division of Computer Science and Engineering, Sun Moon University

요 약

As IT technology developed, storage media also developed. Among them, USB, which is a removable storage medium, is used not only to have several per person but also to work in various companies. Users store valuable and confidential data within USB. As time went on, the need for security increased. In order to solve this security problem, USB has been introduced to allow users to access internal files by inputting ID and password by embedding a security program in USB. However, the method of storing ID and Password inside is low confidentiality and high risk of information leakage. To solve these problems, we propose a 2 factor authentication system using Radius server in addition to login authentication. The proposed system not only improves the authenticity of the device, but also reduces the risk of infringement of personal information when lost. It also encrypts internal files to increase the confidentiality of internal information.

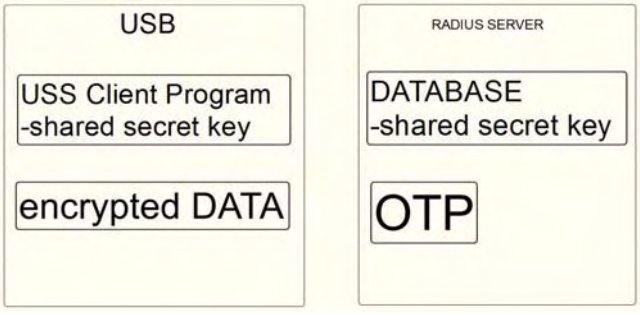
1. 서론

컴퓨터가 발전함에 따라 이동식 저장장치도 빠르게 발전하였다. 그 중 USB드라이브는 엄청나게 높은 보급률을 자랑하고 있다. 개인 사용자들은 USB드라이브 내부에 개인의 소장 자료, 금융거래에 필요한 공인인증서를 저장하여 사용한다. 기업 사용자들은 업무에 필요한 파일 및 기업 내부 보안 파일들을 보관하여 사용한다. 하지만 USB 및 기타 이동장치에는 보안성이 낮다. 그러므로 USB를 잃어버리거나 컴퓨터에 연결하고 나면 누구나 USB에 접근이 가능하고, 정보 유출의 위험성을 내포하고 있다. 또한 USB를 잃어버리면 해당 USB를 찾는 것은 거의 불가능 하며, 이러한 경우 USB뿐만 아니라 내부의 모든 데이터도 잃어버리게 된다. 이러한 문제점이 화제가 되자 보안기능을 가진 USB가 많이 출시되었다. USB보안 프로그램, 지문인식 USB 등의 여러 가지로 보안성을 띄는 USB가 출시되었다 [1-4]. 하지만 기존의 방식들은 크래킹, 인증장치 파괴등으로 보안성이 쉽게 무너질수 있다.

따라서 본 연구에서는 RADIUS[5 서버와 스마트폰 Google OTP[6를 사용한 2차 추가 인증시스템을 제안하고자 한다. USB 내부 암호 알고리즘으로는 Rjindael algorithm[7] 이용하는 AES를 이용한다. 이것은 파일 단위로 암호화를 한다. 제안시스템은 높아진 기밀성을 통하여 USB분실 시에 내장된 파일들을 보호한다. 기업에서 사용할 시 ,기업 내부의 보안파일들을 보다 안정적으로 보관할 수 있게 된다. 그리고 스마트폰 OTP를 이용하여 USB 접근에 대한 기밀성을 증대하여 본 연구의 타당성을 증명하고자 한다.

2. USS 시스템 구조

본 연구에서 제안하는 시스템의 기본 구성은 다음 (그림 1)[5-6]과 같다.



(그림 1) 시스템 구성도

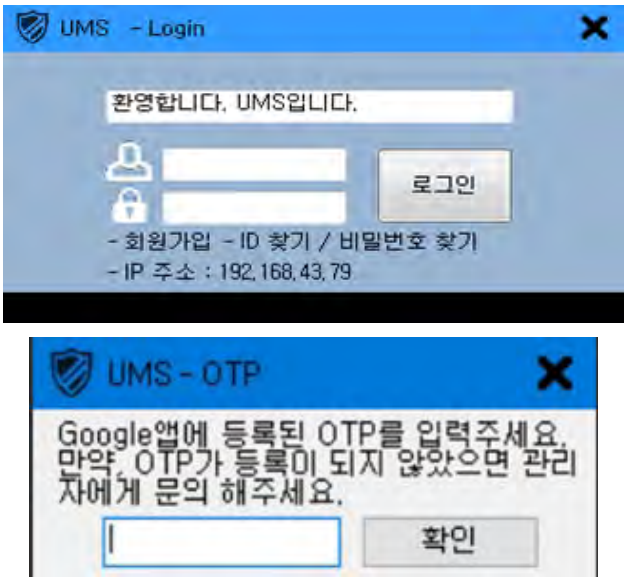
USS 솔루션이 제공하는 기능은 크게 4 가지이다. (그림 1)에서 USB내부에 클라이언트 프로그램을 통해서 인증과정을 수행하여 암호화된 저장 데이터에 접근한다. 인증 시, 1차인증은 기존과 같이 ID/PASSWORD를 입력 받는 로그인 기능으로 수행한다. 기존의 USB 솔루션에서는 보안 프로그램 자체에 사용자의 인증정보를 저장하고 이를 이용하여 인증을 진행한다. 이러한 인증과정은 보안 프로그램이 크래킹 당하면 모든 정보가 노출되어 버리는 단점이 있다. USS에서는 이를 프로그램 자체가 아닌 보안 인증을 적용한 보안서버와의 연동을 통해 인증하는 방식으로 기밀성을 강화한다.

두 번째는 2차인증이다. USS에서는 2차인증 방식으로 OTP를 도입하여 인증절차를 수행한다. 만약 특정 사용자의 ID와 PASSWORD가 유출되었다 하더라도 해당 사용자의 OTP인증을 통하지 않는다면 USB내의 암호화 된 데이터에 접근은 불가능하다. 이러한 2번째 인증 요소를 추가하여 기존의 1개의 인증을 통한 방식보다 높은 기밀성을 제공한다. USS에서는 구글에서 제공하는 Google OTP를 이용한다. 세 번째로는 파일 암/복호화 기능을 제공한다. 2차 인증까지 완료한 사용자는 원하는 파일을 Rijindael algorithm을 이용한 AES로 파일데이터의 암호화 및 복호화 기능을 이용할 수 있다. 기존 솔루션에서는 USB내의 보안영역을 설정한다. 해당 USB의 보안 인증이 무력화 되면 보안영역 내부의 모든 데이터들이 유출되는 위험성이 있다. USS에서는 보안영역이 아닌 파일 단위로 암호화 한다. 그렇기에 USB 내부 데이터가 유출되어도 데이터 자체의 유출은 방지한다. 마지막으로 USB 분실될 시 내부 파일을 모두 삭제하는 기능이 있다. 기존의 솔루션은 인증을 통하여 USB의 특정 보안 영역에 접근을 제한하는 기능만을 제공한다. 하지만 USB자체를 분실할 경우, 해당 USB를 찾는 방법이 없고 USB 내부의 데이터를 분실 및 유출될 위험성이 있다. USS에서는 최소한 USB 분실 시, 내부의 데이터 유출을 막고자 USB내의 모든 암호화 데이터를 삭제하는 기능이 있다. USB가 분실 처리가 되면 내부의 파일들을 이용 할 수 없도록 이를 파기한다.

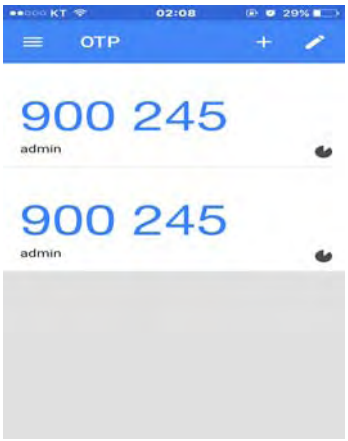
3. 실험 및 분석

(그림 2,3,4)를 보면, 먼저 클라이언트 프로그램을 실행시키고 나서 1차 인증인 사용자 로그인과 2차인증인 OTP 과정을 거쳐서 인증이 완료된다. OTP는 구글 OTP를 이용한다. 인증이 끝난 유저의 클라이언트 프로그램 인터페이스에서 파일의 암/복호화가 가능하게 된다. 이제부터 USS의 각 기능별 구현결과를 확인한다.

클라이언트 1차인증을 수행하면 사용자는 ID와 Password를 입력하면 끝나지만 사용자에게 입력 값을 받으면 해당 인증 값을 RADIUS 서버와의 통신을 통해 유저의 접속 허용 여부를 결정한다.



(그림 2) 클라이언트 1차인증, OTP 2차인증

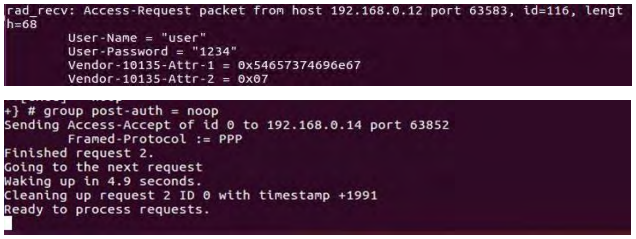


(그림 3) 2차인증 Google OTP

(그림 4) 클라이언트 인증완료, 인터페이스 화면

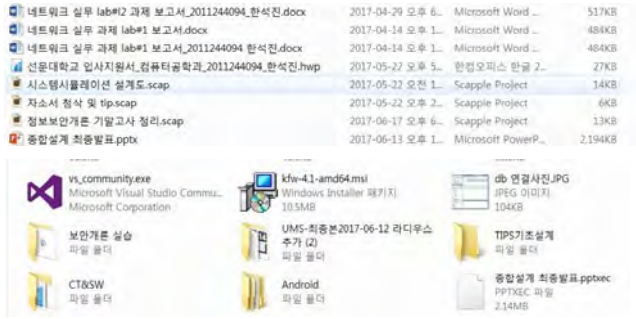
(그림 5)는 RADIUS 서버의 접속을 감지하는 사진으로 사용자 접속 시, 서버와의 DB에서 인증정보를 대조한다. 일치하면 접속을 허용, 없으면 거부한다. 이러한 서버와의 연동으로 기밀성을 높인다.

[시스템 기능별 구현결과]



(그림 5) RADIUS 서버 접속 테스트

그림 6은 인증이 완료된 이후, 사용자의 암/복호화를 완료하고 난 이후의 내용을 보여주고 있다. 사진을 보면 '종합설 최종발표'라는 PPT파일이 있는데 밑의 사진에는 똑같은 제목의 파일에 PPTXEC라는 확장자로 변해있는 파일이 존재한다. 이는 XEC는 암호화 한 이후의 파일의 확장자 제목으로 암호화 이후에는 확장자 이름 뒤에 XEC가 추가되며 AES 암호화가 이루어진다.

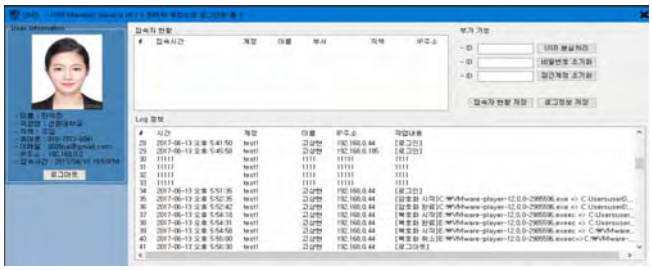


(그림 6) 파일 암/복호화

그림 7은 평문파일의 내용과 암호화 된 파일의 내용을 서로 비교한 내용을 보여주고 있다. 암호화된 파일을 word나 hwp 파일로 변경하면 읽는 것은 가능하지만 사진에서 보듯이, 파일 자체가 암호화가 되어 무슨 내용인지 알 수가 없다.

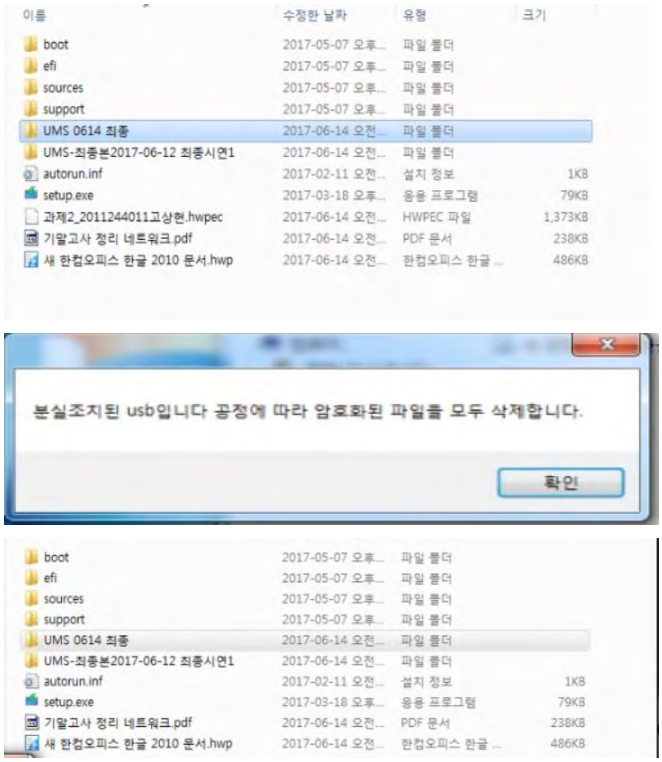
(그림 7) 암호화파일/평문파일

관리자 계정으로 똑같이 2차인증 까지 완료를 하고 나면 그림 8과 같이, 관리자 페이지에 접속한다. 해당 페이지에서는 지금까지 등록된 유저의 로그인 기록, 암/복호화 기록이 전부 기록되어 보여준다. 또한 USB분실처리 및 일반 사용자의 비밀번호 초기화 기능을 이용할 수 있다. 분실처리의 경우 관리자에게 유저가 분실처리 요청을 하게 되면 관리자 계정으로 해당 사용자의 ID를 확인하여 분실처리를 한다.



(그림 8) 관리자 계정 접속/로그확인

마지막으로 분실처리 기능이다. 그림 9처럼 사용자가 관리자에게 USB 분실처리를 요청하면 관리자는 관리자 계정으로 해당 사용자의 USB 분실처리를 한다. 이후, 분실된 USB를 컴퓨터에 연결하고 누군가 클라이언트 프로그램을 실행시킨다. 해당 USB가 분실된 USB라는 것을 RADIUS 서버에서 인식한다. 그리고 USB내의 모든 암호화 파일이 제거한다. 그림을 보면 맨위의 그림에는 XEC확장자의 암호화 파일이 존재하지만 USB 분실처리 이후, 암호화 파일을 전부 삭제한다.



(그림 9) 분실처리

USS는 기존의 암호화 이외에 OTP를 이용한 2차 인증으로 기밀성을 높인다. 그리고 USB 분실대책으로 암호화된

파일을 제거하는 기능을 추가하여 정보유출을 최소화 한다. 이외에도 RADIUS 서버와의 통신을 통해서 인증이 이루어져 기존의 솔루션에 비해 무결성을 증가하였다.

4. 결론

기존에 제안한 프로그램 자체로만 이루어지는 USB 보안 프로그램들은 목적인 기밀성이 상대적으로 낮고 정보 유출의 위험성이 높다. 그렇기에 기업의 경우, 이러한 USB 분실로 인한 정보유출로 피해를 입는 경우가 발생한다. 본 논문에서 제시한 USS 솔루션은 RADIUS 보안 서버와의 연동 및 2 factor authentication을 통한 기밀성을 강화하여 정보유출의 위험성을 줄이고자 하였다. 또한 데이터 자체 암호화를 통하여 정보유출의 위험성을 최소화 하였다.

향후 연구방향으로는 루트킷을 이용한 클라이언트의 자동 인식 및 실행과 내부 망 한정이 아닌 어느 위치에서든 이용 할 수 있도록 구현 하고자 한다. 또한 오프라인 상태에서도 저장영역에 접근이 가능 하도록 구현하여 가용성을 높이고자 한다. USS 솔루션은 프로그램 내부에 AES 암호화에 이용되는 키를 저장한다. 키가 유출 될 가능성이 있기 때문에 보안영역을 추가하여 안전성을 높이도록 한다. Radius Server와 1차 통신과정에서 데이터가 유출 될 가능성이 있다. 통신을 보호하는 수단을 구현할 필요가 있다. 보안 프로그램에서 가장 중요한 것은 암호화 키의 관리이다. 해당 솔루션에서는 이러한 키의 관리수단을 구현하여 기밀성을 높이고자 한다.

참고문헌

[1] Sandisk secureaccess, the resource available at: https://www.sandisk.co.kr/

[2] 지문인식 USB(PD065), the resource available at: http://www.sarotech.com/main/

[3] Sandisk의 secureaccess, windows10의 bitlocker, the resource available at : https://www.sandisk.co.kr/

[4] 고찬, 박연, “Enhancement of Security Function on USB Memory Driver by Reserved Sector Storage Structure Technique”, Journal of the Korea Society for Industrial and Applied Mathemathics-IT series, vol. 9, no. 1, pp. 1-14, 2005.

[5] RADIUS 서버 시스템, the resource available at : http://freeradius.org

[6] The resource available at google website: https://support.google.com/accounts/answer/1066447?hl=ko

[3] Rjindael algorithm, the resource available at: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard