인터넷 인증 및 2 차 생성 파일 암호화를 이용한 소프트웨어 부정 사용 방지 기술[(a)]

박성하*, 채동규**, 김상욱[(b)]**
*한양대학교 컴퓨터공학과
**한양대학교 컴퓨터소프트웨어학과
e-mail:pshda@hanyang.ac.kr

Preventing Unauthorized Software Usage by Internet Authentication and Encryption of Secondary Files

Sung-Ha Park*, Dong-Kyu Chae**, Sang-Wook Kim**
*Dept of Computer Engineering, Hanyang University
**Dept of Computer and Software, Hanyang University

요 약

소프트웨어의 불법 복제 및 인증을 거치지 않은 부정 사용 등이 큰 문제가 되고 있다. 소프트웨어 부정 사용을 막기 위해 usb 인증, 시리얼 키 인증, 서버접속을 통한 인증 등 다양한 방법들이 존재해 왔다. 그러나 이러한 기술들의 우회 방법들이 상당수 공개된 실정이다. 본 연구에서는 보다 강화된 인터넷 인증 기반의 불법 사용 방지 기술을 제안하고자 한다. 뿐만 아니라 소프트웨어를 사용해 만들어 낸 2 차 창작물 또한 보호하는 방법을 제시하고자 한다.

1. 서론

소프트웨어 불법 복제 및 부정 사용은 소프트웨어 산업에 큰 피해를 끼칠 뿐만 아니라 개발자들의 개발 의욕을 낮춘다. 또한 불법적으로 유통되는 소프트웨어는 본래 제작자의 의도와는 다르게 작동하여 소프트웨어의 사용자를 혼란스럽게 하거나 사용자의 시스템에 피해를 주기도 한다. 이러한 문제점을 막기 위해 여러 불법 복제 방지 기술 및 부정사용 방지 기술이 연구되어 왔다. 일정한 규칙을 가진 시리얼 키를 이용하여 인증을 실시하는 방법, CD/DVD 의 자체 복사 방지 기술, USB 장치를 이용한 하드웨어 인증 방법, 컴퓨터의 고유 특성을 이용하여 인증하는 기술, 그리고 서버와 일정한 주기로 통신을 수행하여 인증을 실시하는 방법 등의 방법들이 존재해 왔다 [1].

그러나 시리얼 키를 통한 인증방식은 시리얼 키의 규칙을 파악당할 경우 손쉽게 불법 인증이 가능해지기 때문에 현재는 대부분 다른 인증방식과 같이 사용된다. CD/DVD 의 자체 복사 방지 기술은 하드웨어적으로 복사를 방지할 수 있었지만, 점차 CD/DVD 의 사용률이 낮아지면서 위 방법들의 가치도 하락하게 되었다. 인터넷 인증을 통한 방법의 경우에는 가짜 서버를 구현하거나 서버와 통신하는 패킷을 변조하는 등의 방법으로 간단히 불법 인증이 가능하다. 컴퓨터의 고유 값을 이용한 인증들 또한 고유값들이 쉽게 변조된다는 단점이 있으며, 고유값들 간의 충돌 가능성 또한 존재한다 [1]. 따라서 기존 방법들에 비해 불법 인증이 까다로운 방법이 필요한 실정이다. 뿐만 아니라 기존 방법들의 또 다른 문제는 불법 사용자가 우회 인증을 통해 소프트웨어의 사용권을 획득할 경우 소프트웨어를 사용하여 만들어 낸 2 차 파일들 (예를 들면, MS word 의 경우 사용자가 제작한 word 파일) 까지도 아무 제약 없이 사용할 수 있다는 것이다.

따라서 본 논문에서는 기존 방법들에 비해 강화된 인터넷 인증 기반의 새로운 인증 방법을 제시하고자 한다. 뿐만 아니라 2 차 창작물에 대한 암호화를 통해 부정 사용자는 2 차 창작물에 대한 접근을 원천적으로 차단하는 방법을 제시하고자 한다.

본 논문의 핵심 아이디어를 요약하면 다음과 같다. **(1)** 기존 인터넷 인증 방식은 서버와 클라이언트 사이의 통신을 위, 변조하여 허위로 인증을 수행할 수 있는 단점이 있는데, 본 논문에서는 서버와 클라이언트 통신 사이에 SSL 통신을 수행하여 패킷의 위/변조가 불가능하도록 하여 통신 간 안정성을 확보할 수 있도록 한다. **(2)** 소프트웨어의 실행 파일 생성 시 컴퓨터의 고유값을 포함한 실행 파일이 생성되도록 한다. 본 논문에서는 리눅스 UUID (universally unique identifier)를 고유값으로 사용한다. 그 후 소프트웨어의 매 실행 마다 컴퓨터의 고유값 매칭 및 유효한 사용자인지 확인하는 과정을 거치도록 한다. **(3)** 소프트웨어의 사용으로 얻어지는 2 차 창작을 암호화하여서

(a): 본 연구는 미래창조과학부 및 정보통신산업진흥원의 대학 IT 연구센터 지원사업(NIPA-2013-H0301-13-4009)과 문화체육관광부 및 한국저작권위원회의 2013 년도 저작권기술 개발사업의 연구결과로 수행되었음.
(b): 교신저자.

불법 인증을 거쳐 소프트웨어 사용권을 획득하더라도 암호화를 풀지 못할 경우 접근할 수 없어서 불법 인증의 실익을 얻을 수 없도록 하는 방법을 제안한다.

2. 제안하는 방법

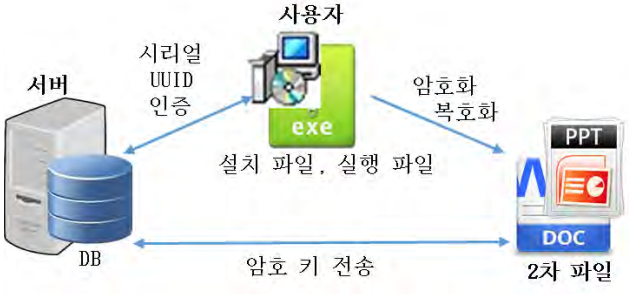

그림 1. 기술 개념도

본 장에서는 제안하는 부정사용 방지 기술에 대해서 순차적으로 설명한다. 먼저 사용자는 설치 프로그램을 실행하여 소프트웨어를 설치하며, 이 과정에서 시리얼 키를 입력한다. 이 때, 설치 프로그램은 입력된 시리얼 키와 함께 사용자 컴퓨터의 UUID 를 서버로 전송한다. 서버는 설치 파일이 보낸 UUID 와 시리얼 키의 유효성과 등록 여부를 DB 에 조회하여 확인한다. 확인이 완료되면 서버는 UUID, 시리얼 키와 더불어 서버에서 지정한 값을 추가하여 SHA256 해시 [2]를 생성한다. SHS256 해시는 2 차 생성 파일의 암호화와 복호화에 사용되는데, 이에 대한 부분은 뒤에서 서술한다. 서버에서 특정 값을 추가하여 해시를 생성하는 이유는 공격자가 사용자의 UUID 와 시리얼 키를 얻었을 경우 이들의 단순 조합만으로 2 차 파일의 암호화에 사용한 해시값을 얻을 수 있기 때문이다.

그 후 서버는 소프트웨어의 실행 파일을 빌드한다. 이 때, 사용자의 UUID 와 시리얼 키가 포함된 SHA256 해시의 checksum 을 실행 파일이 포함하도록 한다. 실행 파일의 생성이 완료되면 서버는 사용자에게 실행 파일을 전송하고, UUID, 시리얼 키, SHA256 해시, SHA256 해시 checksum 을 DB 에 저장한다.

사용자가 서버로부터 전송받은 실행 파일을 실행하게 되면 실행 파일에 포함된 UUID 를 사용자 컴퓨터의 UUID 와 비교하여 일치 여부를 확인한다. 확인이 완료되면 실행 파일에 포함된 checksum 을 계산한 후 서버에 시리얼 키, UUID, checksum 을 질의하여 실행 파일의 무결성과 서버에 전송된 정보가 서버에 보관된 정보와 일치하는지 여부를 확인한다. 이 때 서버가 가지고 있는 값과 사용자로부터 전송 받은 값이 다르다면 소프트웨어는 실행되지 않는다.

다음으로는 2 차 파일의 암호화에 대해서 서술한다. 소프트웨어의 사용으로 만들어진 2 차 생성 파일들을 저장할 때, 소프트웨어는 먼저 암호화의 키 값으로 사용될 SHS256 해시를 서버로부터 받는다. SHA256 해시는 문자열들을 해시 함수로 처리하여 역방향 연산이 불가능하게 만들어 공격자가 쉽게 암호화 시 사용했던 키값을 찾기 어렵게 하여 보안성을 유지하는데 적합하다. 뿐만 아니라 종래의 SHA1 과 달리 아직 공격방법과 충돌 등이 밝혀지지 않았고, 비교적 빠르고 간편한 알고리즘이다. 다음으로 2 차 파일의 암호화 알고리즘은 ARIA 알고리즘 [3]을 사용한다. ARIA 는 국제 표준으로 사용하고 있는 AES 암호화 구조를 가지고 있으며, 소스가 공개되어 있어 구현이 쉽고 아직까지 취약점이 발견되지 않은 알고리즘이기 때문에 본 연구에 적합하다.

ARIA 암호화를 거쳐서 저장된 2 차 생성 파일은 암호화 키를 알지 않는 이상 알 수 없는 형태로 변환되어 부정 사용자는 파일의 내용을 확인할 수다. 저장이 완료된 후에는 SHA256 해시와 checksum 을 서버에 전송한 후 저장 과정을 마무리 한다. 파일을 저장하는 과정에서 블록 암호 알고리즘을 이용해 암호화하기 때문에 암호화 중 소프트웨어가 예기치 않게 작동을 멈추더라도 보안성을 유지할 수 있다.

2 차 파일의 불러오기 과정은 최초 소프트웨어가 실행되었을 때 사용했던 인증을 수행해서 기본적인 소프트웨어 실행 가능 여부를 확인한다. 그 후 불러올 파일의 SHA256 해시 checksum 을 계산한 다음 서버에 질의를 수행하여 이 저장파일의 암호화 시 사용했던 SHA256 해시를 받아온다. SHA256 해시를 받은 후 ARIA 알고리즘을 이용하여 복호화를 수행한다.

이러한 과정을 거친다면 소프트웨어를 부정 사용하여 2 차 창작물이 나오더라도, 서버에 checksum 이 등록되지 않은 파일들은 복호화 할 수 없기 때문에 2 차 창작물을 악의적으로 수정하거나 어떠한 이득을 취할 수 없다. 따라서 부정 사용자는 소프트웨어를 사용하더라도 원하는 동작을 기대하기 어렵다. 또한 모든 인터넷 통신은 인증서를 사용한 SSL 방식으로 통신하여 공격자가 통신을 가로채더라도 내용을 파악할 수 없도록 한다. 또한 클라이언트는 서버의 인증서를 받을 때 서버 인증서의 속성을 확인하여 이 인증서가 접속하고자 하는 서버의 인증서임을 확인한 후 통신을 수행한다. 이를 통해 가짜 서버를 구현하거나 서버와 통신하는 패킷을 변조하는 등의 공격을 받지 않도록 한다 [4].

3. 결론

본 논문에서는 기존의 소프트웨어 부정 사용 방지 방법들에 비해 강화된 인터넷 인증 기반의 새로운 부정 사용 방지 기법을 제안하였다. 핵심 아이디어로서 사용자 컴퓨터의 고유 값이 포함된 실행 파일을 사용하는 것과, 2 차 파일 저장 시 암호화와 불러올 때의 복호화 매커니즘, 서버와 사용자 간의 SSL 통신 등의 방법들을 제시하였다.

참고문헌

[1] Y. Zhang, L. Jin, X. Ye, and D. Chen, “Software Piracy Prevention: Splitting on Client,” *ICST*, 2008

[2] A. Kunhu and H. Al-Ahmad, “Multi Watermarking Algorithm Based on DCT and Hash Functions for Color Satellite Images,” *IEEE IIT*, pp. 30-35, 2013

[3] A. Biryukov, C.D. Canniére and J. Lano, “Security and Performance Analysis of ARIA,” *KU Leuven ESAT/SCD-COSIC*, 2004

[4] M. Lee, “Authentication Reinforcement System Using USB Token”, Mastet’s Thesis in Hankyong Univ., 2007.