Software Aging을 이용한 소프트웨어 저작권 보호 시스템

한효영[0], 박복녕, 김태윤
고려대학교 컴퓨터학과
{takeitez[0], happy, tykim}@netlab.korea.ac.kr

Protection System for Software copy using Software Aging

Hyo-Young Han[0] Bok-Nyong Park Tai-Yun Kim
Dept. of Computer Science & Engineering, Korea University

요 약

소프트웨어 환경에서 업데이트는 버그를 수정하거나 보안패치를 하거나 새로운 기능을 추가시킨다. 그러나 기존의 업데이트는 사용자 인증을 고려하지 않고 단지 cd-key 만으로 사용자 인증을 하거나 웹 사이트에서 자유롭게 업데이트를 할 수 있다. 따라서 본 논문에서는 소프트웨어 환경에서 업데이트시 사용자와 서비스 제공자 사이에 패스워드 기반의 인증 방식을 적용하고, DRM(Digital Rights Management)에서 라이센스 개념을 도입하여 불법 사용자의 접근을 차단할 수 있는 방안을 제시한다.

1. 서 론

소프트웨어 업데이트를 위한 전통적 방식은 익명의 사용자가 웹 사이트에 접속하여 다운받는 방식과 소프트웨어를 설치하면 제약사항 없이 자동으로 업데이트를 받는 방식이었다. 이러한 방식은 불법 사용자들이 쉽게 접근하여 사용하기 쉽다. 또한 Software Aging[1]을 이용한 방식은 불법 사용자가 업데이트에 제약을 받았더라도 소프트웨어를 계속 사용하는데 제약이 없다.

본 논문에서는 소프트웨어 업데이트하기 전에 사용자 인증을 빠르고 비용이 적게 드는 패스워드 기반 인증 방식을 사용하여 업데이트마다 다른 패스워드를 사용하여 패스워드의 재사용을 막아 불법 접근을 차단한다. 또한 DRM[2]의 라이센스에 기초하여 소프트웨어의 사용권을 제한하고 사용자의 편리한 환경을 제공하기 위한 시스템을 제안한다.

본 논문의 구성은 다음과 같다. 2 장에서 기존의 소프트웨어 저작권 보호 방법, Software Aging 기법과 패스워드 기반 인증 프로토콜을 설명한다. 3장에서는 소프트웨어 업데이트에서 요구사항을 고찰하고 제안한 시스템의 시스템 구성과 인증 및 업데이트에 대하여 설명한다. 4장에서는 제안한 시스템의 성능을 기존의 시스템과 비교하며 마지막으로 5장에서는 결론 및 향후 연구 과제를 제시한다.

2. 관련연구

본 장에서는 기존의 소프트웨어 저작권 보호 방법, Software Aging 기법과 패스워드 기반 인증 프로토콜에 대해 설명한다.

2.1 기존의 소프트웨어 저작권 보호 방법

- 등록번호 확인방식: 소프트웨어 설치할 때 소프트웨어 ID를 입력하여 확인하는 방식으로 관리비용이 많이 든다.
- 시간제한 방식: 소프트웨어 나오기 전에 베타 버전에 사용하는 방식으로 사용 유효기간을 적용한 방식으로 크랙으로 인해 사용 제한을 주지 못한다.
- Scramble 방식: CD를 통해 배포되는 소프트웨어에서 사용하게 된다. CD안에 소프트웨어를 사용할 수 있는 키가 내장되어 있어서 실행할 때 마다 CD를 삽입해야 한다. 그러나 최근 Virtual CD가 등장해서 실효성을 거두지 못하고 있다.
- Dongle: 소프트웨어의 정품임을 포함하는 별도의 저장장치를 통해 저장한 뒤 소프트웨어 구동시 이를 확인하는 방식으로 장치보안이 어렵다.

2.2 Software Aging

Software Aging 설계의 기본 목표는 업데이트를 이용하여 소프트웨어 불법 사용자 및 배포자를 막는 것이다. 소프트웨어는 저장되는 모든 파일을 대칭키로 암호화 한다. 그 키는 주기적으로 업데이트를 할 때마다 분배자로부터 키를 받게 된다. 그러나 예전의 키가 저장되지 않게 하기 위해 다음과 같은 일방향 해쉬 함수를 사용하여 업데이트마다 받는 키를 계산한다.

$$K_t = f(K_{t+1}) \qquad (t = n-1,\ n-2, \cdots, 1,\ 0)$$

(t: time interval $t+1$: next time interval)

사용자는 해쉬 체인을 생성하기 위해 처음 할당받은 K_{t+1}을 선택하고 해쉬를 이용하여 주기적으로 새로운 키를 생성할 수 있다. 그래서 불법 사용자는 업데이트를

하지 못하면 최근에 만들어진 파일을 사용할 수 없는 것이다. 그러나 업데이트하기 이전의 소프트웨어는 계속 사용할 수 있기 때문에 사용을 완벽히 차단하지 못하는 단점을 드러내고 있다.[1]

2.3 일회용 패스워드 시스템

패스워드 인증을 위해 S/Key One-Time Password[3]는 eavesdropping와 reply attack으로부터 안전한 인증을 제공한다. 가장 큰 특징은 패스워드 정보는 어떠한 호스트에 보관되지 않는다. 사용자는 ID를 이용하여 서버에 로그인 시도를 한다. 서버는 초기 공유 값(seed)과 해쉬할 횟수(n)를 선택해서 사용자에게 전송한다. 사용자는 자신의 패스워드와 seed 값을 해쉬 함수를 이용하여 n회 해쉬(Pn)한다. 사용자는 ID와 Pn을 전송한다. 서버는 저장되어 있는 사용자 패스워드와 seed값을 계산하여 수신한 값과 일치하면 사용자를 인증한다. 그러나 매 접속 때 마다 패스워드를 바꾸는 장점이 있으나 사용횟수가 제한되어 있고 해쉬 함수에 안전성의 큰 비중을 차지하고 있다.

3. 제안한 시스템 설계

본 장에서 안전한 소프트웨어 환경을 만들기 위해 패스워드 기반 인증 프로토콜 설계와 라이센스를 이용한 소프트웨어 사용권 제한에 대해 설명한다.

3.1 요구사항

안전하고 효율적인 소프트웨어 환경을 구현하기 위해서는 다음과 같은 요구조건들을 만족시켜야 한다.

- 신속한 인증을 통한 업데이트
- 패스워드 공격의 안전성
- 라이센스를 통한 사용권 제어

위와 같은 특징들을 만족시키려면 네트웍 기반에서 사용되는 시스템에서 메시지 교환 횟수와 계산량이 적어야 한다.

3.2 시스템 구성

제안한 시스템은 인터넷을 통한 ESD(Electronic Software Distribution)[4]를 통하여 소프트웨어를 구매하고 라이센스를 받아 사용하는 방식으로 분배자는 사용자의 CPU ID를 가지고 있고 여러 사용자들이 공모하여 하나의 소프트웨어를 구입하지 않음을 가정한다. 그림 1은 제안한 시스템의 업데이트 과정을 나타낸다.

3.3 인증 및 라이센스 업데이트

사용자는 ESD 방식을 이용하여 일정한 지불과정을 거친 후 분배자로부터 소프트웨어를 다운받고 Clearing house로부터 소프트웨어의 라이센스를 다운로드 받는다. 그리고 사용자는 자동으로 주기적인 업데이트를 실시한다. 다음 그림2는 업데이트를 하기 위한 사용자 인증 프로토콜이다.

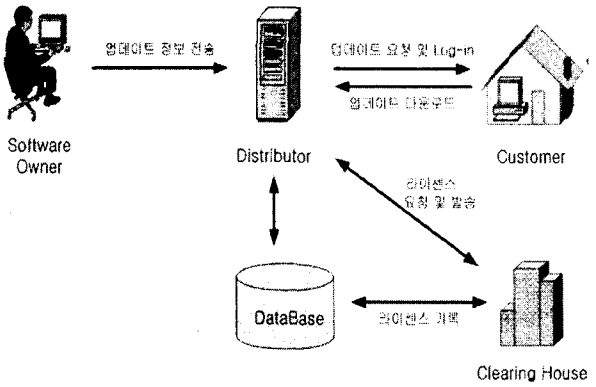

그림 1 제안한 시스템의 업데이트 과정

표 1 제안된 프로토콜의 표기법

·ID : 소프트웨어 식별자
·pwd : 사용자 패스워드
·$seed$: 초기 공유 값
·T_C, T_D: Time stamp 값
·R_C, R_D : Ramdom number
·CID : 사용자의 CPU ID
·$H()$: 일방향 해쉬 함수
·Pn : H^n(pwd||seed)
·$E_K(A)$: 메시지 A를 키 K로 암호화 하는 알고리즘

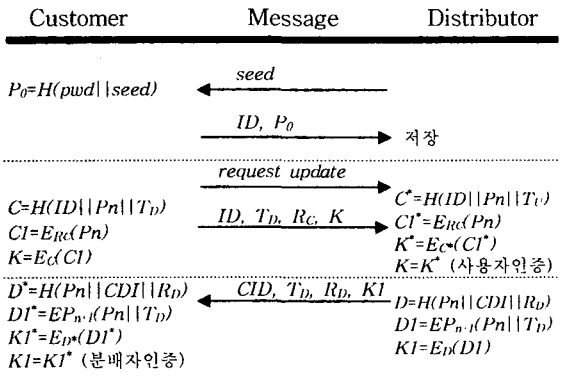

그림 2 사용자 인증을 위한 프로토콜

초기 설정으로 분배자는 안전한 채널을 통해 $seed$값을 보낸다. 사용자는 패스워드와 $seed$값을 해쉬 함수로 연산하여 P_0을 만들어 ID와 함께 보낸다.

사용자는 업데이트를 요구한 후 사용자 ID, 일회용 패스워드 P_n, 사용자 타임스탬프 T_C을 해쉬 함수를 연산하여 C를 만든다. 일회용 패스워드 P_n을 사용자 임의값 R_C로 암호화 하여 $C1$을 만든다. 사용자 인증값으로 쓰는 K는 $C1$을 C로 암호화 하여 생성한다. 그리고 ID, T_D, R_C, K값을 분배자에게 전달한다. 분배자는 사용자 연산과 동일한 방법으로 계산, 비교하여 사용자 인증을 한다.

사용자 인증을 마친 후 분배자는 일회용 패스워드 P_n과 사용자 *CPU ID*와 분배자 임의값 R_D를 해쉬 연산하여 D를 생성하고, 일회용 패스워드 Pn과 분배자 타임스탬프 T_D를 다음에 사용할 일회용 패스워드 P_{n+1}로 암호화 하여 D1을 생성한다. 분배자 인증값 $K1$은 $D1$을 D로 암호화 생성한다. 그리고 CID, T_D, R_D, $K1$을 사용자에게 전달하여 분배자 인증과정을 마친다.

인증과정을 마친 후 분배자는 Software Aging방식을 이용하여 업데이트 과정을 수행한다. 주기적으로 사용자 업데이트 데이터와 소프트웨어를 읽고 쓸 수 있는 키를 분배한다.

소프트웨어는 라이센스에 의해 사용 기간이 제한된다. 소프트웨어는 업데이트할 때 라이센스 정보를 갱신한다. 그리고 라이센스는 암호화 된다. 그림 3은 라이센스 암호화 하는 과정이다.

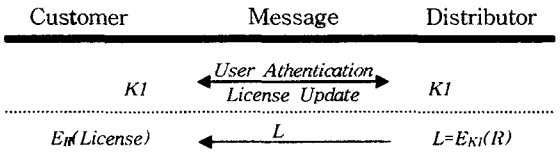

그림 3 라이센스 암호화 과정

인증할 때 서로 공유된 키($K1$)를 이용하여 임의값(R)을 암호화 하여 전송한다. 사용자 소프트웨어에서 라이센스를 R로 암호화 한다. 주기적으로 사용자 라이센스를 갱신하기 위해 업데이트를 해야한다. 그래서 [1]에서의 문제점인 불법 사용자의 업데이트 이전의 소프트웨어 사용을 차단할 수 있다.

4. 제안한 시스템의 성능비교

본 장에서는 제안한 시스템과 기존의 시스템간의 성능을 비교 분석한다.

표 2 제안한 시스템과 기존의 시스템과의 성능 비교

시스템 / 항목	기존 워드프로세서	기존 백신	제안한 시스템
업데이트 방식	웹 사이트	자동	자동
업데이트 사용자 인증 방식	없음	CD-Key	패스워드 방식
불법 복제 방지	X	X	○
불법 유통 방지	X	△	○
사용권 관리	X	X	○
네트워크 의존도	X	△	○

(○: High, △: Low, X: None)

표 2는 제안한 시스템과 기존의 시스템의 성능을 비교 분석 하였다. 기존의 소프트웨어는 웹 사이트, 자동 업데이트를 통해 제한 없이 사용하였으나 제안한 시스템은 업그레이드를 통한 사용제한 방식으로 신속한 사용자 인증을 통해 정식 사용자를 인증한다. 기존의 시스템은 불법 복제 및 유통이 가능하였으나 제안한 시스템은 주기적인 키 업데이트를 통한 키 교환과 사용자 라이센스 암호화를 통해 사용권을 보호하여 불법 복제 및 유통을 차단할 수 있다.

5. 결론 및 향후 연구 과제

본 논문에서는 패스워드 기반 인증을 사용하여 분배자와 사용자간의 인증을 하고, 사용권을 제한하는 라이센스를 도입하였다. 제안한 인증 프로토콜은 일회용 패스워드 방식을 이용하여 패스워드 공격의 안전성을 보장할 수 있다. 또한 라이센스를 업데이트해서 불법 사용자의 사용을 막아 효과적으로 소프트웨어 저작권 문제를 해결한다. 제안한 시스템은 업데이트의 주기가 짧을수록 소프트웨어의 접근 차단 능력이 높아진다. 그러나 분배자의 시스템 자원을 많이 소비하게 된다. 향후 연구과제는 시스템 자원을 적게 사용하면서 안전하게 소프트웨어 저작권을 보호할 수 있는 연구가 필요하다.

참고 문헌

[1] Markus Jakobsson, Michael Reiter "Discouraging Software Piracy Using Software Aging" Workshop on Security and Privacy in Digital Rights Management, November 2001

[2] J. Dubl, "Digital Rights Management : A Definition", IDC, 2001

[3] N. Haller, "The S/KEY one-time password system", RFC 1760, 1995

[4] Fasoo.com, ESD, White paper, http://www.fasoo.com

[5] W. Diffie and M. Hellman, "New directions in cryptography", IEEE Transactions on *Information Theory*, Vol. 22, No. 6, 1976

[6] B. Schneier, Applied Cryptography, Second Edition, John Wiley & Sons, 1996

[7] NBS FIPS Pub 46-1, "Data Encryption Standard", U.S. Department of Commerce, 1988

[8] N. Asokan, V. Shoup, and M. Waidner, "Optimistic protocols for fair exchange," In ACM Security '96, pp.6-17, 1996

[9] R.L.Rivest, "PayWord and MicroMint: Two simple micropayment schemes", 1996