하드웨어에 종속된 암호키 비밀 분할을 이용한 정보권한관리 시스템

(Information Right Management System using Secret Splitting of Hardware Dependent Encryption Keys)

두 소 영 † 공 은 배 ††
(Soyoung Doo) (Eunbae Kong)

요 약 본 논문의 연구 범위는 디지털 음악과 같은 디지털 정보의 지적재산권 소유자의 권한 보호 방법이다. 그 방안으로, 디지털 정보를 인터넷상에서 분배할 때는 암호화 방법을 사용한다. 정당한 사용자가 이 정보를 이용하기 위해서는 해독 과정이 필요하다. 지적재산권을 보호하기 위해 정당한 사용자라도 해독은 시켜주되 해독에 사용되는 키는 모르게 할 필요가 있다. 왜냐하면, 사용자가 해독에 사용되는 키를 알게 되면, 암호화된 정보를 해독하여 사용자가 마음대로 사용할 수 있게 되기 때문이다. 본 논문에서는 사용자로 하여금 해독에 사용되는 키를 알 수 없게 하기 위해서 비밀분할 프로토콜을 사용하며, 추가적인 안전장치로 키를 생성하는데 하드웨어 식별값(MAC 주소 또는 하드 디스크 일련번호)을 활용하고 있다. 사용자가 정보를 사용하고자 할 경우 사용자에게 미리 전달된 디코더가 사용자 식별값을 하드웨어로부터 직접 읽어 해독키를 만들어 낸다. 지정된 시스템이 아닌 경우 해독키가 제대로 만들어 질 수 없으므로 정보를 사용할 수 없게 된다. 따라서, 디지털 정보의 지적 재산권 소유자의 권한은 보다 안전하게 보호 될 수 있다.

Abstract This paper presents a right management scheme using secret splitting protocol. Right management schemes combat piracy of proprietary data (such as digital music). In these schemes, encryption has been used and it is essential to protect the keys used in encryption. We introduce a new key protection method in which a secret encryption key is generated using both user's hardware-dependent unique information (such as MAC address) and cryptographically secure random bit strings provided by data owner. This scheme prevents piracy by checking hardware-dependent information during rendering and improves the secrecy of the data by individualizing the encryption key for each data.

1. 서 론

인터넷의 대중화는 다양한 정보의 디지털화를 촉진시키고, 정보의 유통을 활성화시키는 계기가 되고 있다. 인터넷을 통한 디지털 정보상품의 유통은 무한한 가치를 창출할 수 있음에도 불구하고 몇 가지 문제로 인해 아직 완전한 서비스가 이루어지지 못하고 있다.

지금까지 인터넷을 통한 모든 서비스들은 개방된 시스템을 기반으로 제공되어 왔다. 이 특징 때문에 모든 정보는 자유롭게 공유되었고 정보에 접근하는데 있어서도 특별한 제약이나 어려움이 없었을 뿐더러 비용도 지불하지 않고 사용할 수 있었다. WWW를 통한 정보상품유통은 그 동안 개방되어온 인터넷에서 그 장이 이루어진다는 점에서 커다란 위협을 받고 있다[1]. 예를 들어, 인쇄된 정보의 경우 복사를 할수록 정보의 질은 떨어지게 되지만 디지털 정보는 그 특성상 손쉽게 복사할 수 있고, 이를 다른 사람들에게 다시 전송할 수 있는데 복사 또는 전달된 정보는 원래의 정보와 질적으로 차이가 없으며 원본에도 아무런 흔적을 남기지 않기 때문이

· 본 연구는 과학기술부/한국과학재단 지정 충남대학교 소프트웨어연구센터의 지원에 의한 것임.
† 비 회 원 : 충남대학교 컴퓨터공학과
sydoo@comeng.chungnam.ac.kr
†† 종신회원 : 충남대학교 컴퓨터공학과 교수
keb@comeng.chungnam.ac.kr
논문접수 : 1999년 6월 22일
심사완료 : 1999년 10월 15일

다. 이렇게 아무런 제지 없이 복사와 분배가 허용된다면 정보 소유자에게 커다란 손해를 미치게 될 것이다.

디지털 정보를 상품으로서 유통시키고자 할 때 소유주의 권한을 보호하기 위한 연구는 IBM[2], Intertrust [3], Zerox[4][5], AT&T[6]등의 기업체뿐만 아니라 Stanford[7], CMU[8]등에서도 활발하게 진행되고 있다. 또한, Pay-TV와 같이 동시에 다수의 사용자에게 디지털 정보를 안전하게 전달하기 위한 브로드캐스팅 암호화 방법[9]과 불법으로 사용되는 정보를 추적하는 방법[10][11][12][13]들도 연구되고 있다.

인터넷을 통한 정보 전달 시 정보를 보호하고자 하는 기존의 방법들은 대부분 정보를 암호화된 형태로 전달하고, 해독키를 권한이 있는 특정 사용자에게만 비밀리에 전달하는 방법을 사용하고 있다. 이 방법은 사용자에게 정보가 전달되기까지의 보호는 훌륭하게 처리할 수 있지만, 전달된 해독키가 사용자에게 쉽게 노출 될 수 있다는 단점을 가진다. 즉, 허가된 사용자가 해독키와 암호화된 정보를 권한이 없는 사용자에게 전달할 경우 정보의 부당한 사용을 막거나 검출하기에는 어려움이 있다. 이 문제를 해결하기 위해서는 해독키를 안전한 곳에 숨기거나, 암호키가 발각되더라도 정당한 사용자가 아닌 경우에는 정보를 사용할 수 없도록 하는 방안이 필요하다.

본 논문에서는 암호키 전달로 인한 위험을 방지하기 위하여 사용자 단말기의 유일한 정보를 암호키 생성에 포함시키는 방법을 사용하였다.

먼저, 단말기의 유일한 정보 (단말기의 하드웨어로부터 얻을 수 있는 고유한 값으로 LAN 카드의 MAC(Media Access Control)주소 또는 하드디스크 일련번호 등)와 정보 제공자가 암호학적으로 비밀스럽게 발생시킨 랜덤값을 조합하여 암호키를 생성하며, 생성된 암호키는 비밀분할 프로토콜[14]에 의해 사용자와 정보 제공자에 의해 분리 저장된다. 그리고, 사용자가 암호화된 정보를 사용하고자 할 경우, 정보를 사용하기 위해 정보 제공자로부터 미리 전달받은 디코더(player또는 viewer)에서는 사용자 하드웨어로부터 식별값을 직접 읽어 들이고 이 값과 디코더 내에 숨겨져 있는 랜덤값을 조합하여 해독키를 만들어 낸다. 디코더에서는 정보가 사용되는 사이사이에 하드웨어 식별값을 읽어 정당한 사용인지를 계속 판단한다. 따라서, 암호키가 사용자에게 알려 지더라도 하드웨어 정보가 암호키에 변수로 사용되므로 허가 받지 않은 단말기에서는 정보의 해독이 불가능하게 된다.

이때, 하나의 단말기에서만 정보를 사용하는 것이 문제가 된다면, 정보를 제공하는 서버가 여러 개의 기계에서 정보 사용이 가능하도록 여유를 주어 같은 사용자에게 할당된 다른 암호키로 암호화된 정보를 사용할 수 있게 하여 이 문제를 해결할 수 있다.

본 논문의 구성은 2장에서 디지털 정보 보호를 위해 현재까지 연구되어온 내용을 정리하고, 문제점을 살펴본다. 3장에서는 비밀분할 프로토콜을 사용한 디지털 정보 권한관리 방법에 대해 설명하고, 4장에서 결론과 향후 연구 내용을 정리한다.

2. 디지털 정보 권한 보호 기술 현황

디지털 정보의 권한을 보호하는 기술은 불법적인 사용을 사전에 예방(Prevention)하는 방법과 불법적인 사용을 검출(Detection)해 내는 사후처리 방법으로 크게 나눌 수 있다. 현재까지 디지털 정보를 보호하기 위해 제안되어온 대표적인 방법들을 살펴보면 다음과 같다.

첫 번째 방법은 디지털 정보가 사용되는 동안 계속 사용자와 정보 제공자가 온라인 상태를 유지하여 정보 사용을 모니터링 하는 방법이다[15]. 디지털 정보 사용자가 정보를 불법적으로 사용하지 못하도록 하기 위해서 정보 제공자는 디지털 정보를 정보 제공자가 제공하는 소프트웨어 봉투(Software Envelope)를 통해서만 사용할 수 있도록 한다. 소프트웨어 봉투에는 제공된 정보를 동작시키거나 보는 기능(play 또는 view)과 정보의 사용을 감시하는 기능을 가지고 있다. 예를 들어 음악 파일을 암호화하거나 특별한 포맷으로 생성하여 사용자에게 전달하고 이때 디코더를 함께 전달한다. 이 경우 사용자는 정보 제공자가 제공한 디코더를 사용해서만 음악을 들을 수 있게 된다. 정보 사용자에게 전달한 디코더는 정보가 허용된 권한 한도 내에서 사용되고 있는가를 감시하여 정보제공자에게 온라인 상태로 보고한다. 정보 제공자는 불법적인 정보의 사용이 감지되는 즉시 소프트웨어 봉투에게 동작을 중지하도록 시킬 것이다. 또한, 정보제공자로부터 응답이 없거나, 원하는 메시지 응답이 전달되지 않으면 소프트웨어 봉투의 동작은 중단된다. 이 방법은 디지털 정보의 불법적인 사용을 막기 위한 예방법이다. 그러나, 대부분의 디지털 정보가 소액이라는 점을 고려한다면, 정보 자체의 비용보다 정보를 사용하기 위해 온라인 상태를 유지하는데 필요한 운용비용이 더 많이 들 수 있다는 단점을 가진다.

두 번째 방법은 사용자가 정보를 제공받기 전, 초기화 단계에서 사용자의 개인 정보(주민등록번호, 신용카드번호, 은행계좌번호, 개인암호화 키 정보... 등)를 정보 제공자에게 전달하는 등록 과정을 수행하게 되는데, 사

용자가 정보 제공자로부터 전달받은 정보를 사용할 때 등록 과정에서 전달했던 개인의 비밀 정보를 입력해야만 정보를 사용할 수 있도록 한다. 사용자가 자신의 정보를 정보 사용의 권한이 없는 다른 사람에게 전달한다면, 자신의 중요한 정보도 함께 알려 주어야 할 것이고, 이러한 정보를 남에게 알려 주는 것은 곤란한 상황에 처할 수도 있게 되므로 정보의 무분별한 배포를 억제할 수 있다. 그러나 이 예방법은 그러한 위험을 감수하면서까지 배포하는 사용자를 막을 수는 없다.

세 번째로 인터넷의 특성상 네트워크를 통해 전달되는 정보는 망에 연결된 사용자들 모두에게 노출되어 있다. 정보를 전달하는 사람이 특정 사용자에게만 해당 정보를 전달하고 싶은 경우 암호화 방법을 사용한다. 암호화 방법은 가장 보편적으로 사용되는 방법이다. 이때 암호화된 디지털 정보와 암호를 풀 수 있는 해독키도 사용자에게 전달해 줘야 하는데, 해독키를 파일 형태로 전달하는 것이 대부분의 시스템에서 사용하는 방법이다. 파일 형태로 전달된 해독키는 사용자에 의해서 쉽게 발견되고, 사용자가 발견된 해독키와 암호화된 정보를 함께 다른 사용자에게 전달한다면, 정보 소유주의 권한을 보호하기 위한 디지털 정보의 암호화 방법은 무의미한 일이 된다. 해독키를 직접 전달할 때 발생하는 위험을 줄이기 위해서 사용되는 방법으로는 해독키를 사용자에게 제공되는 디코더내에 숨겨서 보내는 방법이다. 이렇게 함으로써 사용자는 쉽게 해독키를 발견할 수 없게 되고 이 내용이 소프트웨어 보호 방법에 의해서 사용자의 접근을 막아 준다면 정보는 안전할 수 있다. 그러나 이 방법은 디코더가 사용자에 의존적이라는 문제점을 가진다. 즉, 한 사용자에게 전달되는 모든 정보는 하나의 암호키를 사용하게 되고 암호키를 바꾸기 위해서는 디코더를 바꿔야 하는 번거로움이 있다.

네 번째 방법은 디지털 정보 내에 정보의 소유주를 표시하는 정보를 사용자가 알아 볼 수 없고 고칠 수 없게 넣어두는 방법이다[16][17][18][19]. 이 방법은 원래 그림이나 문서의 저작권을 표시하기 위해서 사용되어온 방법이다. 불건전한 사용이 의심되는 경우에 의문을 증명하기 위한 수단으로 사용되는 수동적인 방법으로 워터마크(watermark)와 핑거프린팅(fingerprinting) 기법이 있다. 워터마크는 디지털 정보에 정보 소유주를 표시하는 특별한 내용을 숨겨서 전달한 후 정보의 부정한 사용이 발견되면 이 정보를 검출해 냄으로써 정보 소유주임을 증명하는 방법이다. 이때 숨겨지는 워터마크는 정보 사용자에게 보이지 않아야 하고, 제거할 수 없어야 하며, 정보가 편집된 형태로 부당하게 사용될 수도 있으므로 중요한 부분에 영구적으로 포함될 수 있어야 한다. 핑거프린팅 기법은 디지털 정보마다 다른 내용을 숨겨두고 구매한 사용자가 누구인지를 식별할 수 있게 함으로써 구입한 정보를 다른 사람에게 전달하지 못하도록 한다. 이 방법은 불건전한 사용자가 검색된 경우 정보의 소유주를 증명하는데 사용되는 사후 처리 방법이다.

앞에 소개한 기존의 디지털 정보 보호 방법들 중에서 첫 번째, 두 번째 그리고 세 번째 방법은 정보 제공자가 사용자에게 소프트웨어 봉투나 디코더와 같은 특별한 소프트웨어를 전달하게 된다. 이 소프트웨어는 디지털 정보를 보여주거나 실행시키는 것뿐만 아니라 암호화되어 있는 정보를 해독하고, 정보의 사용이 정당한지 판단하는 기능을 수행한다.

디지털 정보가 가장 보편적으로 사용되는 단말기인 PC(Personal Computer)는 기본적으로 사용자가 자신의 모든 정보에 쉽게 접근할 수 있고, 모든 정보가 공개되어 있어서 정보를 수정하고 관찰할 수 있는 특징을 가진다. 이러한 개방성은 사용자에게 전달된 정보 제공자의 디지털 정보나 디코더와 같은 소프트웨어가 PC 소유자의 통제하에 놓이게 된다는 것을 뜻한다. 정보 제공자가 전달한 소프트웨어의 동작이 사용자에 의해서 추적, 변경되는 것이 가능하다면 사용자는 해독키를 쉽게 찾을 수 있고, 인증 절차를 위조할 가능성도 있다. 따라서, 컴퓨터 내에 사용되는 특별한 소프트웨어의 처리 절차를 사용자가 임의로 변경하거나 관찰 할 수 없도록 하는 소프트웨어 보호 방법이 필요하다. 소프트웨어 보호 방법은 크게 하드웨어를 사용하는 방법과 소프트웨어 방법으로 구분된다.

하드웨어를 사용한 보호방법[20][21]은 특별한 하드웨어를 제공하고 제공된 하드웨어 안에 메모리를 두어 해독키와 해독된 정보를 저장함으로써 사용자가 해독키나 해독된 정보를 직접 접근할 수 없도록 하는 방법이다. 이 방법은 처리 속도와 안전성은 매우 높지만, 시스템의 구조를 변경시키거나 특수한 하드웨어를 제작하여야 하므로 별도의 비용이 필요하다. 때문에 저가의 정보를 판매할 경우에 이 방법은 적합하지 않으며 하드웨어가 사용자에게 보급되기까지의 시간도 필요하다. [21]에서 제안된 내용은 프로그램이 저장되어 있는 기억 장소로부터 프로그램을 꺼내기 위한 명령어를 해독하는 속도를 개선하기 위한 파이프라인(Pipeline) 구조와 전체 프로그램을 사용자가 접근 할 수 없는 기억장소로 읽은 후에 해독해서 실행시키는 캐쉬(Cache) 구조를 소개하고 있다. 소프트웨어가 수행되면서 메모리에 접근하고

처리하는 과정을 추적하여 보면 프로그램의 동작을 유추할 수 있게 된다. 이러한 추적으로부터 프로그램의 동작을 감추기 위해서 의미 없는 더미코드를 추가하고 비순차적인 처리를 수행함으로써 소프트웨어의 보호를 꾀하는 방법도 제안되고 있다.

소프트웨어를 사용한 보호방법[22]은 하드웨어 방법에 비해서 안전성이 떨어 질 수 있으나 저가 디지털 정보와 같은 상품을 유통하기 위해서는 효율적으로 동작할 수 있는 장점을 가진다. 소프트웨어로 프로그램을 보호할 경우에는 숨기고자 하는 프로그램의 정보를 시간과 공간적으로 분산시키고, 메모리 구조를 복잡화 시켜서 동작중인 메모리를 검색하는 것이 불가능하도록 한다. 또한 프로그램의 동작 주기에 차이를 두어서 혼란을 일으키고, 소프트웨어의 각 인스턴스가 일련번호와 같은 유일한 코드를 프로그램이 인스톨될 때 저장하여 가지도록 하는 방법도 사용된다. 이러한 기능을 통합하여 프로그램을 구성함으로써 프로그램이 동작할 때 같은 일을 수행하면서도 그 절차는 매번 다른 처리 방식으로 수행되어 프로그램의 추적을 어렵게 만들 수 있다.

본 논문에서는 정보 제공자가 사전에 제공하는 디코더와 같은 소프트웨어가 소프트웨어 보호 방법에 의해 안전하게 동작한다고 가정한다. 따라서 사용자는 디코더에서 수행되는 동작이나 절차를 쉽게 추적할 수 없으며 해독된 정보는 동작(play 또는 view)후 바로 삭제되므로 비밀스러운 정보를 사용자가 찾아내는 것은 매우 어렵다고 가정한다.

3. 비밀분할 프로토콜을 사용한 권한관리 기술

본 논문에서는 디지털 정보를 보호하는 가장 보편적인 방법인 정보를 암호화해서 전달하는 방법을 개선하여 보다 안전한 디지털 정보 소유주의 권한 보호 방법을 제시하고자 한다. 우선, 파일 형태로 해독키가 제공되는 문제를 해결하기 위해서 사용자에게 키를 전달하지 않고, 사용자가 이미 가지고 있는 정보로부터 해독키를 만들어내는 방법을 사용하였다. 사용자마다 모두 다른 암호키를 사용하기 위해서 각 사용자를 구분할 수 있는 값이 필요한데 이 값은 사용자 기계로부터 얻을 수 있는 유일한 정보를 이용한다.

3.1 사용자 식별 번호

단말기를 식별하기 위해서 제안되는 방법으로는 컴퓨터 하드웨어 모듈이 가지는 불변하는 값이 이용되고 있다. 대표적으로 사용되는 하드웨어로는 LAN 카드와 하드디스크를 예로 들 수 있다.

LAN 카드의 MAC 주소는 LAN 카드가 생산될 때 부여되는 번호로 하나의 LAN 카드가 가지는 MAC 주소(6 바이트)는 전 세계에서 유일한 값이다. 이 값은 응용프로그램에서 쉽게 읽어올 수 있고 LAN 카드를 교체하지 않는 한 불변하므로 통신 프로토콜에서 뿐 아니라 컴퓨터를 식별하는데도 유용하게 사용되고 있다.

식별자	값
MAC주소	00-10-5A-61-49-53
하드디스크 일련번호	13EA-1E57

그림 1 하드웨어 식별값의 예

하드디스크 일련번호는 포맷을 하기 전까지는 늘 유일한 값을 가지고 있게 된다. 포맷을 할 때마다 변화되는 값이므로 LAN 카드보다는 덜 안정적이나 하드디스크를 포맷 한 후에 프로그램들을 다시 인스톨해야 하므로 인스톨 수행 시 하드디스크 일련번호를 점검하여 새로운 값으로 교체하거나 재등록하는 절차를 추가함으로써 프로그램에서 컴퓨터를 식별하는 값으로 사용하기에 문제가 없다.

컴퓨터에 사용자의 개별성을 부여하는 방법은 하드웨어정보가 정보 제공자에게 등록된 컴퓨터에서만 사용할 수 있다는 제약이 따른다. 그러나 이 문제는 응용 프로그램에서 한 사용자가 여러 단말기의 하드웨어 정보들을 등록할 수 있는 여유를 줌으로써 충분히 해결 가능하다.

MAC 주소나 하드디스크 일련번호만으로 암호키를 생성하는 경우 한 사용자에게 전달되는 모든 정보가 하나의 암호키를 사용하게 된다. 하나의 암호키가 알려지면 모든 정보가 노출될 수 있으므로, 이러한 위험을 줄이기 위해서 키를 분산 저장하는 비밀분할 프로토콜을 사용한다.

3.2 비밀분할 프로토콜

어떤 사람이 정보를 한 사람에게 전부 알려줄 경우 비밀이 유지되지 못할 것을 우려하여 여러 조각으로 나누어 각 사람에게 한 조각씩 보관하도록 한다. 나누어진 조각은 각각의 조각만으로는 아무 의미가 없고 모든 조각이 모여야만 의미 있는 정보가 되는데 이것이 바로 비밀분할 프로토콜[14-3.6]이다. 예를 들어 어떤 음식점에 유명한 요리 비법을 가진 주인이 주방에서 일하는 사람들에게 조리법을 조금씩 나누어서 알려 주었다고 하자. 이 사람들 모두가 모여서 음식을 만들게 된다면 음식은 훌륭한 맛을 내겠지만, 어느 한 사람이 가진 정보로는 음식을 제대로 만들 수 없게 된다. 이 프로토콜은 사용자의 수가 늘어나도 동일한 방법으로 적용이 가

능하다. 단, 참여자 들 중 누군가가 메시지를 잃어버리거나 참여하지 못하는 상황이 발생한다면 메시지를 재생할 수 없다는 단점이 있다.

하드웨어에서 직접 얻어오는 값만으로 암호키를 생성할 경우 모든 정보에 하나의 암호키가 적용된다는 단점을 해결하기 위해서 본 논문에서는 정보 제공자가 선택한 암호학적으로 비밀스럽게 발생(예를 들면 Blum-Blum-Shub generator[23])하는 랜덤값 (R: 랜덤비트열)과 사용자 하드웨어 식별값의 연산을 통해 암호키(Ek)를 생성한다. 사용자에게 전달되는 정보는 암호키로 암호화되는데, 각 정보마다 랜덤값이 변화되어 모든 정보가 다른 암호키를 사용하게 된다.

이때 사용자는 하드웨어 식별값을 가지고 있고, 정보 제공자는 R을 가지고 있다. 해독키(Dk)를 생성하기 위해서 정보 제공자는 R을 사용자에게 알려주어야 한다. 사용자가 정보의 해독키(Dk)를 구하는 방법은 다음식과 같다.

$$D_K = D_{V_k}(E_{V_k}(R)) \oplus \text{하드웨어식별자}$$

3.3 비밀분할을 사용한 권한관리 방법

디지털 정보 권한 관리 시스템의 동작은 다음과 같이 이루어진다.

(1) 초기화 단계

사용자들은 정보 제공자에게 연결하여 등록 절차를 수행한다. 이 과정에서 사용자는 사용자의 MAC 주소, 하드디스크 일련번호, 그리고 신용카드 번호와 같은 사용자 정보를 정보 제공자에게 전달한다.

정보 제공자는 MAC주소나 하드디스크 일련번호와 같은 사용자 식별값과 이와 동일한 크기의 R을 생성하여, 연산을 통해 암호키(Ek)를 만들어 내고, 이 값을 전달된 사용자 정보와 함께 데이타 베이스에 저장하여 관리한다. 정보 제공자가 가지고 있는 데이타 베이스에는 사용자 정보, 제공된 정보, R 등이 기록되어 있다. 사용자에게는 정보를 재생할 수 있는 디코더를 전송한다. 사용자가 정보를 사용하기 위해서는 비밀분할 프로토콜에 의해서 분산되어 있는 해독키 정보를 모두 제공해 주어야 한다. 사용자 식별자인 MAC 주소나 하드디스크 일련번호는 사용자가 알고 있으므로 R을 비밀리에 전달하여 주는 것이 필요하다. R을 전달할 때 사용할 수 있는 방법은 다음과 같다.

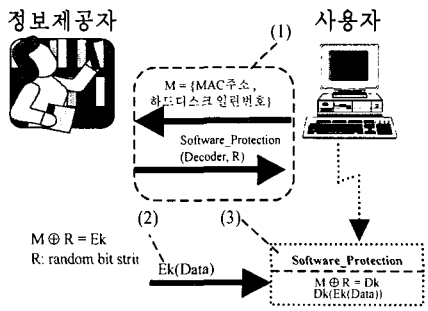

그림 2 사용자 식별값을 사용하는 암호화 방법의 수행 절차

● 키 분배(Key distribution) 방법

R을 디코더에 숨겨서 전달하는 방법으로 디코더가 소프트웨어 보호 방법에 의해서 보호되므로 사용자의 메모리에서도 안전한 곳에 저장될 수 있고 사용자는 이 값을 찾아 낼 수 없다. 이 방법은 모든 사용자에게 R을 각각 할당하여 주어야 하고 이 값을 기억해야 하는 단점이 있으나 R이 밝혀진다 해도 하나의 디코더에 한하게 되므로 그렇게 심각한 문제가 발생하지는 않는다는 장점을 가진다. 또한, 암호화 절차가 공개되어도 키가 안전하게 관리되므로 암호 시스템에는 문제가 발생하지 않는다. 그러나, 한 사용자에게 전달되는 모든 정보는 동일한 암호키로 암호화 되므로 다른 암호키를 사용하고 싶은 경우 디코더를 다시 배포해야 하는 번거로움이 있다.

● 함수 분배(Function distribution) 방법

이 방법은 R을 직접 전달하지 않고 R을 계산해 낼 수 있는 함수를 전달하는 방법이다. MAC 주소나 하드디스크 일련번호와 함께 해독키를 만들어내는 R을 정보 제공자와 정보사용자가 동일한 값을 만들어 낼 수 있는 함수를 통해서 생성한다. 여기서 함수의 동작은 소프트웨어 보호 방법으로 사용자가 추적할 수 없으므로 안전하게 R을 구할 수 있다. 모든 사용자에게 동일한 디코더를 전달해도 되고, 동일한 사용자라 하여도 정보마다 다른 암호키를 사용하여 정보를 전달한다는 장점을 가진다. 함수의 처리내용이 발견될 경우를 대비하여 주기적으로 갱신된 디코더로 바꾸어야 하는 단점이 있다.

● 키 협상(Key agreement) 방법

사용자와 정보의 보호 정도에 따라 암호키를 사용자와 정보 제공자가 협상하여 생성해 내는 방법을 사용한다. 암호키 생성 방법은 잘 알려져 있는 키 교환 알고리듬들을 사용할 수 있는데, 사용자와 정보 제공자간에 Diffe-Hellman키 교환[23]방법을 수행하면 다음과 같다.

① 사용자는 랜덤 값 a_U를 선택한다. $(0 \le a_U \le p-2)$

② 사용자는 $\alpha^{a_U} \bmod p$를 계산한 후 이것을 정보 제

공자에게 보낸다.

③ 정보 제공자는 랜덤 값 a_V를 선택한다. $(0 \le a_V \le p-2)$

정보 제공자는 $\alpha^{a_V} \bmod p$를 계산한 후 이것을 사용자에게 보낸다.

④ 사용자는 $K=(\alpha^{a_V})^{a_U} \bmod p$를 계산한다.

⑤ 정보제공자는 $K=(\alpha^{a_U})^{a_V} \bmod p$를 계산한다.

이 방법은 키를 매우 안전하게 생성할 수 있고, 키 생성 방식이나 키가 노출되어도 큰 위험이 없다. 다만, 절차가 다른 방법에 비해 좀 복잡하고 처리비용이 많이 든다.

위의 세 가지 방법들은 정보의 중요도와 시스템의 특성에 따라 효율적인 방법을 결정하여 선택적으로 운용될 수 있다.

(2) 정보 전달 단계

사용자는 정보 제공자가 제공하는 정보를 검색한 후, 원하는 정보를 선택하고, 지불 절차를 수행한다. 정보 제공자는 지불이 정상적으로 수행되면 정보를 사용자에게 암호화된 형태로 전달하게 된다. 이때 초기화 단계에서 만들어진 암호키를 사용한다. 암호화된 정보와 데이타 베이스에 보관되어 있는 R을 함께 사용자에게 전달한다. R의 전달 방법은 단계(1)에서 제시한 방법 중 정보의 중요도에 따라 선택한다. 사용자마다 하드웨어 식별값이 다르기 때문에 다른 암호키가 사용되며, 정보마다 R도 계속 변화하도록 하여 각각의 정보는 모두 다른 암호키가 사용된다. 정보 제공자의 데이타 베이스에는 사용자에게 보내지는 새로운 정보 식별번호와 R이 추가 관리된다.

(3) 정보 사용 단계

사용자 하드웨어로부터 읽어온 하드웨어 식별값과 정보 제공자로부터 전달된 랜덤 값 R을 사용하여 해독키를 생성한다. 암호화된 디지털 정보는 소프트웨어 보호방법에 의해서 보호되는 디코더에서 해독키를 사용하여 해독되므로, 사용직전에 해독되고, 디코더를 통해 사용된 직후 지워진다.

만약 권한이 없는 다른 컴퓨터에서 암호화된 정보를 사용하게 된다면 디코더가 읽어 들인 사용자의 하드웨어 정보가 다르기 때문에 올바른 해독키(Dk)를 얻을 수 없고 원래의 디지털 정보를 복원할 수도 없게 된다.

디지털 정보를 비밀분할을 이용한 암호키로 암호화하는 방법이 기존의 방법과 비교할 때 가지는 장점을 정리하면 다음과 같다.

첫째는 현재까지 사용되어온 암호화 키 전달 방식처럼 정보를 전달할 때마다 해독키를 비밀스럽게 별도로 전달할 필요가 없다. 둘째는 해독키가 밝혀져도 다른 사용자 시스템에서의 재생을 막을 수 있다. 본 논문에서 제안한 비밀분할에 의한 암호키 생성방법은 해독키의 일부만이 정보와 함께 전달되고, 나머지 정보는 사용자 기계의 고유한 값으로써 하드웨어에서 직접 읽어 들이므로 키가 발각되어 다른 사용자에게 전달된다 하더라도 하드웨어 식별값이 암호키를 만들 때 사용된 값과 달라서 정보를 재생할 수 없다. 셋째, 각 사용자별로 또 전달되는 모든 메시지마다 암호키가 모두 다르므로 키가 발견될 위험이 줄어 안전성이 뛰어나다.

4. 결 론

정보 소유주의 권한을 보호하는 기존의 방법 중에서 가장 보편적으로 사용되는 방법은 정보를 암호화하여 전달하고, 해독키를 특정 사용자에게만 전달하는 것이다. 그런데, 이 방법은 사용자가 권한이 없는 사용자에게 해독키와 암호화된 정보를 함께 제공할 경우 암호화 처리가 무의미하게 된다.

본 논문에서는 암호화에 사용된 키를 사용자에게 알리지 않고 전달할 수 있는 방법과 정보를 사용하는 중에 정당한 사용자 인지를 확인하는 방법을 제안하였다. 암호키를 사용자의 유일한 정보인 하드웨어 식별값과 정보 제공자가 만들어낸 랜덤값을 이용하여 만들고 이 값을 비밀분할 프로토콜을 사용하여 분리 저장한다. 분리 저장된 해독키는 디지털 정보가 사용될 때 하드웨어로부터 직접 읽혀진 식별값과 정보 제공자가 미리 전달한 디코더에 들어있는 랜덤값이 결합하여 만들어지게 된다. 여기서 하드웨어 정보는 정보 사용 중에 지속적으로 검사되어 사용자임을 증명하는 역할을 하게 되고, 랜덤값은 정보마다 다른 암호키를 생성시키는 기능을 한다. 즉, 기계를 식별하여 디지털 정보 사용 권한이 주어지고, 정보마다 새로운 암호키를 사용할 수 있으므로 안전성이 뛰어나 정보 소유주의 권한을 최대한 보호할 수 있게 된다.

본 논문에서 제안한 방법은 첫째, 해독키를 별도로 전달할 필요가 없다는 것과 둘째, 해독키가 발견되더라도 사용자에 의해 불법 복사된 디지털 정보를 다른 사용자 시스템에서 재생할 수 없다는 것 그리고, 각 사용자별로 또 전달되는 메시지마다 암호키가 모두 달라서 해독키가 발견되더라도 하나의 정보에 한하므로 위험성을 줄일 수 있다는 장점을 가진다.

제안된 방법의 문제점으로는 허가된 사용자라 하더라도 정보를 사용할 수 있는 단말기가 제한적이라는 것인

데 이것은 허가된 사용자가 사용하는 단말기를 다수 개로 지정하는 방법으로 해결 가능하다.

본 논문에서는 사용자 시스템에 전달되어 정보를 해독하는 디코더 프로그램을 소프트웨어 보호방법에 의해 보호된다는 가정을 하고 있다. 사용자가 프로그램의 동작을 추적하거나 비밀스럽게 숨겨둔 내용을 찾아내려고 할 때 이를 막기 위한 다양한 방법에 대한 연구가 향후 지속적으로 필요하다.

참 고 문 헌

[1] 두소영, 공은배, "인터넷 상에서 전달되는 디지털 정보의 권한 관리 기술", 제10회 정보보호와 암호에 관한 학술대회, pp.19-25, 1998.

[2] H.M.Gladney, "Access Control for Large Collections," ACM Transactions on Information Systems, vol.15, No.2, pp.154-194, 1997.

[3] InterTrust, "Securing the Content, Not the Wire, for Information," URL: http://www.intertrust.com/technology/stc.html.

[4] Mark Stefik, "Letting Loose the Light: Igniting Commerce in Electronic Publication," Xerox Palo Alto Research Center, Palo Alto, CA, 1995.

[5] Mark Stefik, "The Digital Property Rights Language. Manual and Tutorial. Version 1.02," Xerox Palo Alto Research Center, Palo Alto, CA, 1996.

[6] R. Martin Roscheisen, "A Network-Centric Design For Relationship-Based Rights Management," Stanford Univ., URL:http://pcd.stanford.edu/rmr/dissertation.pdf, 1997.

[7] A.K.Choudhury, N.F.Maxemchuk and S.Paul, H.G.Schulzrinne, "Copyright Protection for Electronic Publishing over Computer Networks," IEEE Network Magazine, pp.12-20, 1994.

[8] Bennet Yee, "Using Secure Coprocessors," CMU-CS-94-149, 1994.

[9] A.Fiat, M.Naor,"Broadcast Encryption," CRYPTO'93, pp.480-491. 1994.

[10] B.Chor, A.Fiat and M.Naor,"Tracing Traitors," CRYPTO'94 LNCS 839, pp.257-270, 1994.

[11] Birgit Pfitzman, "Trails of Traced Traitors," Information Hiding, LNCS 1174, pp.49-64, 1996.

[12] Cynthia Dwork, Jeffrey Lotspiech and Moni Naor, "Digital Signets: Self-Enforcing Protection Digital Information," 28th Symposium on Theory of Computing, pp.489-498, 1996.

[13] Kaoru Kurosawa and Yvo Desmedt, "Optimum Traitor Tracing and Asymmetric Schemes," EUROCRYPT '98, LNCS 1403, pp.145-157, 1998.

[14] Bruce Schneier, Applied Cryptography, pp.758, WILEY, 1996.

[15] Gary N.Griswold, "Method for Protecting Copyright on Networks," http://www.cni.org/docs/ ima.ip-workshop/Griswold.html, 1993.

[16] Birgit Pfitzmann and Matthias Schunter, "Asymmetric Fingerprinting," EUROCRYPT'96, LNCS 1070, pp.84-95, 1996.

[17] Birgit Pfitzmann and Matthias Schunter, "Asymmetric Fingerprinting for Larger Collusions," 4th ACM Conf. On Computer and Communications Security, pp.151-160, 1997.

[18] Dan Boneh and James Shaw, "Collusion-Secure Fingerprinting for Digital Data," CRYPTO'95, LNCS 963, pp.452-465, 1995.

[19] Nasir Memon and Ping Wah Wong, "Protecting Digital Media Content," Communications of the ACM, Vol.41, No.7, pp.35-43, 1998.

[20] Amir Herzberg and Shlomit S. Pinter, "Public Protection of Software," ACM Transactions on Computer Systems," Vol5, No.4, pp.371-393, 1987.

[21] Oded Goldreich and Rafail Ostrovsky, "Software Protection and Simulation on Oblivious RAMs," Journal of the ACM, Vol.43, No.3, pp.431-473, 1996.

[22] David Aucsmith, "Tamper resistant software: An Implementation, In Ross Anderson, editor Information Hiding," First International Workshop, May/June 1996, LNCS 1174, pp.317-333, 1996.

[23] Douglas R. Stinson, Cryptography Theory and Practice, pp.434, CRC Press, 1995.

두 소 영

1998년 ~ 현재 충남대학교 컴퓨터공학과 박사과정. 1994년 ~ 1997년 고등기술연구원. 1994년 충남대학교 컴퓨터공학과 석사. 1992년 군산대학교 정보통신공학과 졸업. 관심분야는 암호프로토콜, 디지털정보권한보호, 전자상거래, 네트워크보안.

공 은 배

1996년 ~ 현재 충남대학교 컴퓨터공학과 부교수. 1995년 Oregon State Univ. 전산학 박사. 1978년 ~ 1981년 서울대학교 계산통계학과 석사. 1974년 ~ 1978년 서울대학교 계산통계학과 졸업. 관심분야 인공지능, 기계학습, 전자상거래, 암호학.