The Journal of The Institute of Internet, Broadcasting and Communication (IIBC)
Vol. 17, No. 4, pp.35-41, Aug. 31, 2017. pISSN 2289-0238, eISSN 2289-0246

https://doi.org/10.7236/JIIBC.2017.17.4.35
JIIBC 2017-4-5

비밀 분산 및 스테가노그래피를 이용한 개인 키 보관 기법

Private Key Management Scheme Using Secret Sharing and Steganography

이재흥*

Jaeheung Lee*

요 약 본 논문은 개인 키 저장을 위해, 개인 키를 (k, n) 비밀 분산 기법을 통하여 n개의 조각으로 나눈 후, 스테가노그래피 기술을 사용하여 각 조각을 서로 다른 사진 파일에 저장하는 기법을 제안한다. 사용자는 전체 n개의 사진 파일 중 k개가 어디에 있는지만 기억하고 있으면 개인 키를 복구할 수 있으며, 시스템에 저장된 수많은 사진 파일들 사이에 적절하게 개인 키 조각들을 숨겨놓으면 공격자는 어디에 개인 키가 저장되어 있는지 찾아내기가 힘들다. 사용자는 n개의 사진 파일 중 k개의 위치만 기억하면 개인 키를 복구 할 수 있기 때문에 사용자 편의성도 높은 편이고, 공격자가 k-1개의 사진 파일을 찾아내더라도 개인 키를 복구하는 것은 불가능하기 때문에 안전성도 보장된다.

Abstract This paper introduces a new method for storing a private key. This method can be achieved by dividing the private key into "n" pieces by a (k, n) secret sharing method, and then storing each piece into photo files utilizing a steganography method. In this way, a user can restore a private key as long as he can remember the locations of "k" photos among the entire photo files. Attackers, meanwhile, will find it extremely difficult to extract the private key if a user has hidden the pieces of the private key into numerous photo files stored in the system. It also provides a high degree of user convenience, as the user can restore the private key from his memory of k positions among n photo files. Coupled with this, a certain level of security can be guaranteed because the attacker cannot restore a private key, even if he knows k-1 photo file locations.

Key Words : Private Key, Secret Sharing Scheme, Steganography, Certificate, Bitcoin

Ⅰ. 서 론

정보통신 기술의 발전으로 인터넷 뱅킹이나 쇼핑과 같이 기존 오프라인에서 이루어지던 많은 활동들을 PC나 스마트폰을 통해 수행할 수 있게 되었다. 이를 공격하여 불법적인 이득을 취하려는 시도 역시 증가하고 있는데, 이를 막기 위해 개발된 다양한 기법들 중 공개 키 기반구조(Public Key Infrastructure, PKI) 방식의 공인인증서가 현재 가장 널리 사용되고 있다.

공인인증서는 전자 서명 검증에 필요한 공개 키에 소유자 정보를 추가하여 만든 일종의 전자 신분증으로 공개 키가 해당 소유자의 것이 맞다는 것을 인증기관이 보증한다.[1] 이를 통해 인터넷에서 거래 당사자 간의 신원 확인 및 부인 방지가 가능해지며, 따라서 공인인증서는 전자상거래 활성화에 있어 필수적인 요소이다.

특정 공인인증서의 공개 키로 검증 가능한 전자 서명

*정회원, 대전대학교 정보보안학과
접수일자: 2017년 7월 25일, 수정완료: 2017년 8월 10일
게재확정일자: 2017년 8월 11일

Received: 25 July, 2017 / Revised: 10 August, 2017 / Accepted: 11 August, 2017
*Corresponding Author: leejh@dju.kr
Department of Computer & Information Security, Daejeon University, Korea

을 생성하기 위해서는 이와 쌍을 이루는 개인 키가 필요하다. 공격자가 누군가의 개인 키를 알아내면 다양한 전자상거래에서 그 사람으로 위장할 수 있기 때문에 개인 키를 어떻게 관리하는가가 공인인증서 기반의 보안 시스템의 안전성을 구성하는 핵심 요소가 된다.

개인 키 관리가 중요한 것은 현재 온라인 가상 화폐로 가장 많이 사용되는 비트코인에서도 마찬가지이다.[2] 비트코인은 분산화 구조로 이루어진 블록체인 기반의 가상화폐로, 자신이 가진 비트코인을 사용하기 위해서는 해당하는 개인 키가 필요하다. 비트코인에서는 가상 화폐를 사용하기 위한 개인 키들을 전자 지갑에 넣어 관리하는데 이를 탈취해 부정 인출하는 사례가 이미 여러 차례 발생하였다.[3] 따라서 개인 키를 어떻게 관리하는가는 비트코인 시스템에 있어서도 매우 중요하다.

개인 키를 관리하는 가장 간단한 방법은 파일 형태로 내용 그대로 저장하는 것이다. 하지만 이 방법은 개인 키를 저장한 시스템이 해킹될 경우 키를 그대로 노출하게 된다. 또한 시스템이 고장날 경우를 대비해 키 백업이 필요한데 외부 시스템에 키 백업을 수행할 경우 이를 어떻게 신뢰할 것인가에 대한 문제도 있다.

현재 공인인증서나 비트코인 전자 지갑에서 개인 키 보관을 위해 가장 많이 사용하는 방법은 개인 키를 사용자 비밀번호에서 추출한 키로 암호화하여 저장하는 것이다. 공인인증서에서 개인 키는 보통 PKCS #5에 정의된 사용자 비밀번호 기반 암호화 기법[4]으로 암호화하여 PKCS #8 형식의 파일로 저장하여 보관한다.[5] 비트코인 전자 지갑에서도 개인 키 보관을 위해 패스프레이즈를 사용해 개인 키를 암호화하여 저장할 수 있다.[6, 7] 하지만 이 방법은 개인 키를 저장한 시스템이 해킹될 경우 공격자가 무작위 대입 공격이나 사전 공격을 수행할 수 있게 된다. 사용자들이 편의를 위해 무작위 대입 공격이나 사전 공격에 약한 비밀번호를 사용하는 경우가 많아 이 경우 개인 키가 쉽게 노출된다. 또한 공격자가 포털 사이트와 같은 다른 사이트를 해킹해 얻은 비밀번호를 통해 유추하거나 사용자 PC에 키보드 해킹 프로그램을 몰래 설치해 개인 키 암호화에 사용된 비밀번호를 얻어 개인 키를 찾아내는 것도 가능하다.

본 논문에서는 개인 키 저장을 위해, 개인 키를 Shamir의 (k, n) 비밀 분산 기법[8]을 통하여 n개의 조각으로 나눈 후, 스테가노그래피 기술을 사용하여 각 조각을 서로 다른 사진 파일에 저장하는 기법을 제안한다. 사용자는 전체 n개의 사진 파일 중 k개가 어디에 있는지만 기억하고 있으면 개인 키를 복구할 수 있으며, 시스템에 저장된 수많은 사진 파일들 사이에 적절하게 개인 키 조각들을 숨겨놓으면 공격자는 어디에 개인 키가 저장되어 있는지 찾아내기가 힘들다. 사용자는 n개의 사진 파일 중 k개의 위치만 기억하면 개인 키를 복구 할 수 있기 때문에 사용자 편의성도 높은 편이고, 공격자가 k-1개의 사진 파일을 찾아내더라도 개인 키를 복구하는 것은 불가능하기 때문에 안전성도 어느 정도 보장된다.

또한 이 기법은 클라우드 환경에 개인 키를 저장하기에 적합하다. 현재 클라우드에 저장된 파일들 중 상당수가 사진 파일인데, 이 사진들 사이에 개인 키 조각들을 숨겨놓으면 공격자가 클라우드에 저장된 사진 파일들을 다 얻을 수 있다 하더라도 그 안에 개인 키 정보가 있다는 것을 알아채기 힘들며, 알아도 수많은 사진들 중 어디에 있는지 찾기가 쉽지 않다.

본 논문은 다음과 같이 구성된다. 2장에서는 관련 연구로 공인인증서와 비트코인 환경에서 개인 키를 어떻게 저장하는지 알아보고, 비밀 분산 기법과 스테가노그래피 기법에 대해 설명한다. 3장에서 제안 기법에 대해 자세히 설명하고, 4장에서 제안 기법의 타당성을 증명하기 위해 다른 개인 키 저장 방법들과 비교 분석하며, 5장에서 결론을 맺는다.

II. 관련 연구

1. 공인인증서 개인 키 저장 기법

인터넷에서 거래 당사자 간의 신원 확인 및 부인 방지를 위해 공인인증서가 널리 사용되고 있다. 특정 공인인증서의 공개 키로 검증 가능한 전자 서명을 생성하기 위해서는 이와 쌍을 이루는 개인 키가 필요하다. 공격자가 누군가의 개인 키를 알아내면 다양한 전자상거래에서 그 사람으로 위장할 수 있기 때문에 개인 키를 어떻게 관리하는가가 공인인증서 기반의 보안 시스템의 안전성을 구성하는 핵심 요소가 된다.

현재 공인인증서에서 개인 키 보관을 위해 가장 많이 사용하는 방법이 PKCS #5에 정의된 사용자 비밀번호 기반 암호화 기법이다.[4] 구체적으로 PKCS #5에 정의된 PBKDF1에 해시 함수[9, 10]로 SHA-1을 사용하여 사용자 비밀번호를 키와 초기 벡터로 변환하고, PBES1

(Password-Based Encryption Scheme)에 암호 알고리즘으로 SEED를 사용하여 개인 키 정보를 암호화하여 저장한다. 이 때 저장 형식은 PKCS #8을 따른다.[5]

이 방법은 공격자의 손에 암호화된 개인 키 파일이 들어갈 경우, 사용자들이 편의를 위해 단순한 비밀번호를 사용하는 경우가 많아 공격자가 무작위 대입 공격이나 사전 공격을 수행하여 쉽게 개인 키를 얻을 수 있다는 단점이 있다. 또한 공격자가 포털 사이트와 같은 다른 사이트를 해킹해 얻은 비밀번호를 통해 유추하거나 사용자 PC에 키보드 해킹 프로그램을 몰래 설치해 개인 키 암호화에 사용된 비밀번호를 얻는 것도 가능하다.

2. 비트코인 개인 키 저장 기법

가상 화폐인 비트코인을 사용하기 위해서는 해당 주소에 대응하는 개인 키가 필요하다.[2, 6, 7] 비트코인에서는 화폐 사용을 위한 개인 키들을 전자 지갑에 넣어 관리하는데 이를 탈취해 부정 인출하는 사례가 이미 여러 차례 발생한 바가 있다.[3] 따라서 개인 키의 안전한 관리는 비트코인 시스템에 있어서 매우 중요하다.

비트코인은 구조화된 파일이나 간단한 데이터베이스 형태로 구현된 지갑 안에 개인 키들을 저장한다.[7] 비트코인 지갑은 개인 키를 생성하는 방식에 따라 비결정적 지갑과 결정적 지갑으로 나뉜다. 비결정적 지갑은 다양한 개인 키들을 무작위적으로 생성하고, 결정적 지갑은 공통 종자에서 일방 해시 함수를 적용하여 개인 키를 얻는다. 비결정적 지갑은 많은 키를 보유해야 하기 때문에 그다지 권장되지 않으며, HD 지갑으로 불리는 결정적 지갑이 많이 사용된다.

비트코인에서는 개인 키를 안전하게 보관하기 위해 패스프레이즈를 이용해 개인 키를 암호화한다.[6, 7] 이 때 암호 알고리즘으로 AES를 사용하며, Base64Check를 통해 인코딩을 수행한다. 이렇게 암호화된 개인 키는 종이 지갑의 형태로 인쇄하여 콜드 스토리지라고 알려진 오프라인 지갑에 보관할 수도 있다.

3. 비밀 분산 기법

1979년 Shamir는 비밀 데이터 D를 n개의 조각으로 나누어 그 중 k개 이상의 조각이 있으면 D를 쉽게 계산할 수 있지만, k-1개 이하의 조각이 있을 경우 D에 관한 정보를 전혀 얻을 수 없도록 한 (k, n) 비밀 분산 기법을 제안하였다.[8]

이 기법은 다음과 같이 동작한다.

① 비밀 데이터 D를 수로 변환한다. 이를 a_0라 하자.

② k-1개의 난수 a_1, a_2, a_3, ..., a_{k-1}을 생성한다.

③ n개의 조각 D_1, D_2, D_3, ..., D_n 값을 다음과 같이 계산하여 분산 저장한다.

$$q(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_{k-1} x^{k-1} \quad (1)$$

$$D_1 = q(1), D_2 = q(2), D_3 = q(3), \ldots, D_n = q(n) \quad (2)$$

q(x)는 k-1차 다항식이므로 k개의 서로 다른 (i, D_i) 값을 알면 q(x)를 계산할 수 있다. 그럼 a_0 값도 구할 수 있고(a_0=q(0)이므로), 이를 통해 비밀 데이터 D를 복구하는 것이 가능하다. 하지만 공격자가 k-1개의 서로 다른 (i, D_i) 값을 알더라도 a_0 값을 계산하는 것은 불가능하고, 따라서 비밀 데이터 D의 내용도 알 수 없다.

이 기법은 실수 상에서만 적용 가능한 것이 아니라 p가 소수일 때, 유한체인 Z_p에서도 적용 가능하다. 이를 이용하여 비밀 데이터의 크기가 클 경우 일정 비트, 또는 바이트 단위로 나눠서 처리하는 것도 가능하다.

4. 스테가노그래피 기법

암호화 기법의 경우 공격자가 내용은 알 수 없지만, 비밀 메시지가 있다는 사실은 알 수 있는 반면 스테가노그래피 기법은 비밀 메시지의 존재 자체를 숨긴다. 이를 위해 이미 존재하는 다양한 형태의 데이터(예. 사진, 동영상, 문서, 프로그램)에 약간의 변형을 가해 메시지를 삽입한다. 이를 통해 많은 메시지를 넣을 수는 없지만 존재 여부를 숨길 수 있어 많은 관심을 받고 있다.

사진 또는 그림 파일에 비밀 메시지를 숨기기 위해 사용하는 스테가노그래피 기법 중 가장 유명한 것이 LSB 삽입 기법이다.[11] 이는 각 픽셀을 이루는 값들 중 가장 영향을 조금 미치는 LSB 값을 메시지의 내용에 따라 변경함으로써 기존 파일과 거의 같게 보이면서 메시지의 존재를 숨긴다. 이 방법은 비밀 정보를 손쉽게 삽입할 수 있고, 종류에 따라 많은 내용을 넣을 수도 있지만 이미지 변환에 약하다는 단점이 있다. 이 기법은 직접적으로 각 픽셀에 대한 RGB 값들을 인코딩하는 방식을 사용하는 BMP나 PNG 파일에 바로 적용 가능하다.

JPEG 파일의 경우에도 DCT 수행 후 양자화 된 계수의 LSB 값에 메시지를 삽입하는 방법으로 LSB 삽입 기법을 사용할 수 있다.[12] Jsteg이 이러한 방식을 사용하는 대표적인 예이다.[13] Jsteg은 사진 파일의 약 12.8%에 해

당하는 큰 삽입량을 가지며, DCT 수행 후 양자화 된 계수 값이 0 또는 1인 경우에는 삽입 대상에서 제외하여 이미지의 변화를 최소화한다.

Ⅲ. 제안 기법

본 장에서는 개인 키 저장을 위해, 개인 키를 (k, n) 비밀 분산 기법을 통하여 n개의 조각으로 나눈 후, 스테가노그래피 기술을 사용하여 각 조각을 서로 다른 사진 파일에 저장하는 기법을 제안한다. 제안 기법은 계산량을 줄이기 위해 (k, n) 비밀 분산 기법을 개인 키 전체에 바로 적용하지 않고, 일정 크기로 나눠서 처리한다. 표 1은 제안 기법에서 사용하는 변수 및 기호를 나타낸다.

표 1. 제안 기법에서 사용하는 변수 및 기호
Table 1. Variables and Notations

이름	설명
k	개인 키를 복구하기 위해 필요한 최소 조각 수
n	개인 키 정보가 보관된 전체 조각 수
|X|	X의 바이트 단위 크기
||	문자열 결합(concatenation)
M	개인 키 정보
D_j	비밀 분산 기법을 적용하여 n개로 나누어진 개인 키 정보 j는 $1 \le j \le n$인 정수
mLen	비밀 분산 기법 적용을 위한 메시지 크기 (바이트 단위)
t	비밀 분산 기법 적용을 위해 나누어진 메시지 개수
p	비밀 분산 기법이 계산되는 기반 유한체 Z_p의 크기 소수(prime number)이며, |p| ≥ mLen + 1이어야 함

1. 개인 키 분산

본 절에서는 개인 키를 (k, n) 비밀 분산 기법을 통하여 n개의 조각으로 나누는 과정을 상세히 기술한다. 앞에서 설명한 바와 같이 계산량을 줄이기 위해 (k, n) 비밀 분산 기법을 개인 키 전체에 바로 적용하지 않고, 일정 바이트 단위로 나눠서 처리한다. 이를 위한 구체적인 과정은 다음과 같다.

① 개인 키 정보를 M이라 하고, |M|을 M의 바이트 단위 크기라 하자. M은 개인 키 자체일수도 있고, 사용자 비밀번호 기반 암호화를 적용한 개인 키일 수도 있다. |M|이 mLen의 배수가 아닐 경우, mLen의 배수가 되도록 뒤에 패딩을 붙여 M'을 만든다. 이 때 패딩 값은 패딩의 크기와 같은 mLen – (|M| % mLen)이다. |M|이 mLen의 배수일 경우에도 mLen 번만큼 mLen 값으로 패딩한다.

$$
\begin{aligned}
M' &= M \,||\, 01 \\
&\quad (|M| \ \%\ mLen = mLen - 1\text{일 때}) \\
&= M \,||\, 02 \,||\, 02 \\
&\quad (|M| \ \%\ mLen = mLen - 2\text{일 때}) \\
&\qquad \cdots \\
&= M \,||\, (mLen\text{-}1) \,||\, \ldots \,||\, (mLen - 1) \\
&\quad (|M| \ \%\ mLen = 1\text{일 때}) \\
&= M \,||\, mLen \,||\, \ldots \,||\, mLen \\
&\quad (|M| \ \%\ mLen = 0\text{일 때})
\end{aligned}
\tag{3}
$$

② M'을 mLen 바이트 단위로 나누어 $M_1, M_2, \ldots, M_t$를 만든다. (t = |M'| / mLen)

$$M' = M_1 \,||\, M_2 \,||\, \ldots \,||\, M_t \tag{4}$$

③ 각 M_i ($1 \le i \le t$)에 대해 아래와 같이 (k, n) 비밀 분산 기법을 적용하여 $M_{i1}, M_{i2}, M_{i3}, \ldots, M_{in}$ 값을 계산한다.

- k – 1개의 난수 $a_{i1}, a_{i2}, a_{i3}, \ldots, a_{i(k-1)}$을 생성한다. ($a_{ij}$는 $0 \le a_{ij} < p$ 인 정수)
- $M_{i1}, M_{i2}, M_{i3}, \ldots, M_{in}$ 값을 다음과 같이 계산하여 각각 저장한다.

$$q_i(x) = M_i + a_{i1}x + \ldots + a_{i(k-1)}x^{k-1} \bmod p \tag{5}$$

$$M_{i1} = q_i(1), M_{i2} = q_i(2), \ldots, M_{in} = q_i(n) \tag{6}$$

④ 인덱스 값과 각 조각 정보를 취합하여 $D_1, D_2, D_3, \ldots, D_n$을 만든다. 여기서 M_{ij}는 $0 \le M_{ij} < p$인 정수이므로 |p| 바이트를 사용하여 인코딩하고 인덱스 값은 1 바이트를 사용한다. 따라서 $|D_j| = 1 + t \times |p| = 1 + \frac{|M'|}{mLen} \times |p|$이다.

$$D_j = j \,||\, M_{1j} \,||\, M_{2j} \,||\, M_{3j} \,||\, \ldots \,||\, M_{tj} \ (1 \le j \le n) \tag{7}$$

2. 개인 키 조각 저장

본 절에서는 n개의 조각으로 나누어진 개인 키($D_1, D_2, D_3, \ldots, D_n$)를 스테가노그래피 기술을 사용하여 서로 다른 사진 파일에 저장하는 기법에 대해 설명한다. 다양한 종

류의 사진 파일에 개인 키 조각을 넣을 수 있지만 여기서는 인터넷에서 가장 많이 쓰이는 JPEG 파일을 예로 들어 설명하도록 하겠다. BMP나 PNG 파일의 경우 직접적으로 각각의 픽셀에 대한 RGB 값들을 인코딩하는 방식을 사용할 수 있어 삽입 가능한 메시지 용량이 크고 화질도 좋지만, 파일 크기가 커서 일반적으로 사용되지는 않는다. 따라서 이러한 파일들의 경우 의심을 사기가 쉽다.

JPEG 파일의 경우 양자화를 통한 손실 압축으로 인해 BMP나 PNG 파일에서처럼 각 픽셀 정보에 직접적으로 데이터를 넣는 방식은 사용할 수 없다. 대신 DCT 수행 후 양자화 된 계수의 LSB 값에 메시지를 삽입하는 방법을 주로 사용한다. Jsteg이 이러한 방식을 사용하는 대표적인 예이다.[13] 이 방법을 사용하여 사진 파일에 개인 키 조각을 저장하는 과정을 간략하게 기술하면 다음과 같다.

① 각 조각 D_j $(1 \le j \le n)$ 앞에 길이 정보를 추가해 D_j'을 만든다. 이를 위해 맨 앞 5 비트 공간 L에 다음에 나오는 $|D_j|$의 비트 단위 크기를 넣고, 그 뒤 0에서 31 비트 사이의 공간에 D_j의 바이트 단위 크기인 $|D_j|$ 값을 적은 후 D_j를 붙여 D_j'을 만든다.

$$D_j' = L \,||\, |D_j| \,||\, D_j \quad (1 \le j \le n) \qquad (8)$$

② D_j' $(1 \le j \le n)$을 넣기 위한 n개의 JPEG 사진 파일을 선택한다. 각 사진 파일에 대해 ③~⑤를 수행한다.

③ Huffman 디코딩을 수행하여 압축을 푼다.

④ 양자화된 DCT 계수 값을 순서대로 찾아 이 값이 0 또는 1인 경우는 무시하고, 그 외의 경우 D_j'에서 한 비트씩 데이터를 차례대로 읽어 해당 DCT 계수 값의 LSB를 덮어씌운다.

⑤ 다시 Huffman 인코딩을 수행한다.

이러한 방법으로 n개의 사진 파일에 개인 키 조각을 저장한 후, 이 파일들을 파일 시스템 내부에 적절하게 분산하여 저장하면 된다. 이 때 파일 수정 시간이 개인 키 조각을 저장한 파일들을 찾기 위한 힌트가 될 수 있으므로 적절하게 바꿀 필요가 있다.

3. 개인 키 조각 획득

본 절에서는 개인 키 정보가 들어있는 n개의 사진 파일 중 k개의 사진 파일을 찾아 개인 키 조각을 획득하는 과정을 기술한다.

① 개인 키 복구를 위해 찾은 k개의 사진 파일 각각에 대해 ②~③을 수행한다.

② Huffman 디코딩을 수행해 압축을 푼다.

③ 양자화된 DCT 계수 값을 순서대로 찾아 이 값이 0 또는 1인 경우는 무시하고, 그 외의 경우 LSB에서 한 비트씩 읽어들여 다음과 같이 처리한다.

- 처음 5 비트 정보를 통해 $|D_X|$의 비트 단위 크기를 구한다. (아직 인덱스 값을 모르기 때문에 D_X로 표현)
- 앞에서 구한 비트 단위 크기만큼 더 읽어 D_X의 바이트 단위 크기인 $|D_X|$를 구한다.
- 마지막으로 $|D_X|$ 바이트만큼 더 읽어 D_X를 구한다.

4. 개인 키 계산

본 절에서는 k개의 개인 키 조각으로 개인 키를 복구하는 과정을 기술한다.

① 개인 키 조각 각각의 첫 바이트를 읽어 인덱스 값을 구하고, 나머지 부분은 |p| 바이트 단위의 블록으로 나눈다.

$$D_x = x \,||\, M_{1x} \,||\, M_{2x} \,||\, M_{3x} \,||\, \ldots \,||\, M_{tx} \quad (1 \le x \le n) \qquad (9)$$

② ①에서 나눈 각 블록 별로 k개의 연립방정식을 만들어 부분 개인 키 M_1, M_2, ..., M_t를 계산한다.

③ M_1, M_2, ..., M_t를 다 연결해 M'을 만든다.

$$M' = M_1 \,||\, M_2 \,||\, \ldots \,||\, M_t \qquad (10)$$

④ M'에서 패딩 부분을 제거해 개인 키 M를 구한다.

Ⅳ. 평가

본 장에서는 제안 기법의 타당성을 증명하기 위해 다른 개인 키 저장 기법들과 비교 분석한 결과를 제시한다. 표 2는 보안성, 편의성 및 가용성 관점에서 개인 키 저장 기법들을 평가한 결과이다.

표 2. 개인 키 저장 기법 평가
Table 2. Evaluation of Private Key Management Schemes

개인 키 저장 기법	보안성	편의성	가용성
개인 키 그대로 저장	×	○	△
개인 키 그대로 저장 (백업 수행)	××	○	○
사용자 비밀번호 기반 암호화	○	△	×
제안 기법	○	△	○

(○: 좋음, △: 보통, ×: 나쁨, ××: 아주 나쁨)

개인 키를 그대로 저장하는 기법은 사용하기는 쉽지만 개인 키를 저장한 시스템이 고장나서 키를 잃어버리거나 해킹되어 키가 유출될 수 있는 위험이 있다.

시스템의 고장으로 인한 키 분실을 막기 위해 백업을 수행할 수도 있지만 이 경우 키 유출의 위험은 더 커진다.

개인 키를 사용자 비밀번호에서 추출한 키로 암호화하여 저장하는 방법은 시스템이 해킹된다 하더라도 키가 바로 노출되지는 않지만, 암호화된 개인 키 정보를 바탕으로 공격자가 무작위 대입 공격이나 사전 공격을 수행할 수 있고, 많은 사람들이 기억하기 쉽도록 간단한 또는 의미가 있는 비밀번호를 많이 사용하기 때문에 위험할 수 있다. 또한 사용자가 무작위 대입 공격이나 사전 공격에 강한 비밀번호를 사용할 경우 기억하기가 힘들어 개인 키의 분실 가능성이 그만큼 높아진다.

인지 과학 분야에서의 다양한 실험을 통해 알려진 결과에 따르면, 사람들은 사진 또는 그림을 인지하고 기억하기 위한 매우 거대한 기억 공간을 가지고 있다[14, 15]. 또한 2000년 미국 캘리포니아 버클리 대학교에서 20명을 대상으로 수행한 사용자 연구에 따르면, 사용자 인증 시 표 3에서와 같이 PIN 번호나 사용자 비밀번호를 사용하는 경우보다 랜덤 아트나 사진 포트폴리오를 사용하는 경우 로그인 실패율이 더 낮아짐을 알 수 있다[16].

표 3. 로그인 실패율 비교 (출처: [16])
Table 3. % Failed logins

	PIN	비밀번호	랜덤 아트	사진
로그인 실패율	5%	5%	0	0
로그인 실패율 (일 주일 뒤)	35%	30%	10%	5%

제안 기법에서 사용자는 전체 n개의 사진 파일 중 k개가 어디에 있는지만 기억하고 있으면 개인 키를 복구할 수 있으며, 시스템에 저장된 수많은 사진 파일들 사이에 적절하게 개인 키 조각들을 숨겨놓으면 공격자는 어디에 개인 키가 저장되어 있는지 찾아내기가 힘들다. 사용자는 n개의 사진 파일 중 k개의 위치만 기억하면 개인 키를 복구 할 수 있기 때문에 사용자 편의성 및 가용성이 높은 편이고, 공격자가 k-1개의 사진 파일을 찾아내더라도 개인 키를 복구하는 것은 불가능하기 때문에 안전성도 어느 정도 보장된다.

V. 결 론

본 논문에서는 개인 키 저장을 위해, 개인 키를 (k, n) 비밀 분산 기법을 통하여 n개의 조각으로 나눈 후, 스테가노그래피 기술을 사용하여 각 조각을 서로 다른 사진 파일에 저장하는 기법을 제안하였다. 사용자는 n개의 사진 파일 중 k개의 위치만 기억하면 개인 키를 복구 할 수 있기 때문에 사용자 편의성도 높은 편이고, 공격자가 k-1개의 사진 파일을 찾아내더라도 개인 키를 복구하는 것은 불가능하기 때문에 안전성도 어느 정도 보장된다.

본 논문에서 제안한 기법은 클라우드 환경에 개인 키를 저장하기에도 적합하다. 현재 클라우드에 저장된 파일들 중 상당수가 사진 파일인데, 이 사진들 사이에 개인 키 조각들을 숨겨놓으면 공격자가 클라우드에 저장된 사진 파일들을 다 얻을 수 있다 하더라도 그 안에 개인 키 정보가 있다는 것을 알아채기 힘들며, 알아도 수많은 사진들 중 어디에 있는지 찾기가 쉽지 않다.

앞으로도 본 논문에서 제안한 기법이 사용하고 있는 다양한 매개변수들의 최적 값에 대한 연구를 계속 진행할 예정이다.

References

[1] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, RFC 5280, May (2008).
DOI: https://doi.org/10.17487/RFC5280

[2] S. Nakamoto, Bitcoin: A Peer-to-Peer Electronic Cash System, http://bitcoin.org/bitcoin.pdf, (2009).

[3] T. Moore and C. Nicolas, Beware the middleman: Empirical analysis of Bitcoin-exchange risk, International Conference on Financial Cryptography and Data Security, Springer Berlin Heidelberg, (2013).
DOI: https://doi.org/10.1007/978-3-642-39884-1_3

[4] B. Kaliski, PKCS #5: Password-Based Cryptography Specification Version 2.0, RFC 2898, September (2000).

DOI: https://doi.org/10.17487/RFC2898

[5] B. Kaliski, PKCS #8: Private-Key Information Syntax Specification Version 1.2, RFC 5208, May (2008).
DOI: https://doi.org/10.17487/RFC5208

[6] Mike Caldwell and Aaron Voisine, Passphrase-protected private key, https://github.com/bitcoin/bips/blob/master/bip-0038.mediawiki.

[7] Andreas M. Antonopoulos, Mastering Bitcoin: Unlocking Digital Cryptocurrencies, O'Reilly Media, Inc., (2014).

[8] Adi Shamir, How to share a secret, Communications of the ACM, Volume 22, Issue 11, Pages 612-613, Nov. (1979).
DOI: https://doi.org/10.1145/359168.359176

[9] Hie-Do Kim, A Study on the Secure Double Pipe Hash Function, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 10, No. 6, pp. 201-208, Dec 2010.
DOI: https://doi.org/10.7236/JIIBC.2010.12.201.

[10] Hyung-Kyu Yang, A Fast and Secure Method to Preserve Anonymity in Electronic Voting, The Journal of The Institute of Internet, Broadcasting and Communication, Vol. 14, No. 1, pp. 245-251, Feb 2014.
DOI: https://doi.org/10.7236/JIIBC.2014.02.245.

[11] Jessica Fridrich, Miroslav Goljan, and Rui Du, Detecting LSB steganography in color, and gray-scale images, IEEE multimedia, Volume 8, Issue 4, Pages 22-28, (2001).
DOI: https://doi.org/10.1109/93.959097

[12] Jessica Fridrich, Tomáš Pevný, and Jan Kodovský, Statistically Undetectable JPEG Steganography: Dead Ends, Challenges, and Opportunities, In Proceedings of the 9th workshop on Multimedia & security (MM&Sec '07), ACM, New York, NY, USA, 3-14, (2007).
DOI: https://doi.org/10.1145/1288869.1288872

[13] D. Upham, Jsteg, http://www.securityfocus.com/tools/1434, (1997).

[14] Ralph Norman Haber, How we remember what we see, Scientific American, Volume 222, Issue 5, Pages 103-112, May (1970).

[15] L. Standing, J. Conezio, and R.N. Haber, Perception and memory for pictures: Single-trial learning of 2500 visual stimuli, Psychonomic Science, Volume 19, Issue 2, Pages 73-74, (1970).

[16] R. Dhamija, A. Perrig, Deja Vu: User study using images for authentication, In Ninth Usenix Security Symposium, (2000).

저자 소개

이 재 홍(정회원)

- 2001년 2월 : 서울대학교 컴퓨터공학부 졸업
- 2003년 2월 : 서울대학교 전기, 컴퓨터공학부 석사
- 2013년 2월 : 서울대학교 전기, 컴퓨터공학부 박사
- 2016년 3월 ~ 현재 : 대전대학교 정보보안학과 조교수

<주관심분야 : 암호학, 정보보호, 시스템 보안, 무선 보안>