JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 28 No. 3, pp. 59-66, March 2023
https://doi.org/10.9708/jksci.2023.28.03.059

A Study on the Private Key Backup and Restoration using Biometric Information in Blockchain Environment

Seungjin Han*

*Professor, Dept. of Software Convergence, Kyung-In Women's University, Incheon, Korea

[Abstract]

As research on blockchain applications in various fields is actively increasing, management of private keys that prove users of blockchain has become important. If you lose your private key, you lose all your data. In order to solve this problem, previously, blockchain wallets, private key recovery using partial information, and private key recovery through distributed storage have been proposed. In this paper, we propose a safe private key backup and recovery method using Shamir's Secrete Sharing (SSS) scheme and biometric information, and evaluate its safety. In this paper, we propose a safe private key backup and recovery method using Shamir's Secrete Sharing (SSS) scheme and biometric information, and evaluate its safety against robustness during message exchange, replay attack, man-in-the-middle attack and forgery and tampering attack.

▸ **Key words**: Blockchain, Biometric, Private Key, Repository, Back up, Restoration

[요 약]

다양한 분야에서 블록체인을 적용한 연구가 활발하게 증가함에 따라 블록체인의 사용자를 증명하는 개인키의 관리가 중요하게 되었다. 개인키를 분실하게 되면 본인의 모든 데이터에 대한 권한을 잃게 된다. 이에 대한 문제점을 해결하고자 기존에는 블록체인 지갑, 부분 정보를 이용한 개인키 복구, 분산 저장을 통한 개인키의 복구 등을 제안하였다. 본 논문에서는 Shamir's Secrete Sharing(SSS) 스킴과 생체정보를 이용하여 안전한 개인키의 백업 및 복구 방안을 제안하고 이에 대한 안전성을 메시지 교환시의 견고성, 재생공격, 중간자 공격 및 위변조 공격에 대해 평가한다.

▸ **주제어**: 블록체인, 생체인식, 개인키, 저장소, 백업, 복구

• First Author: Seungjin Han, Corresponding Author: Seungjin Han
*Seungjin Han (softman@kiwu.ac.kr), Dept. of Software Convergence, Kyung-In Women's University
• Received: 2023. 02. 01, Revised: 2023. 03. 14, Accepted: 2023. 03. 14.

http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

I. Introduction

블록체인은 Satoshi Nakamoto가 비트코인이라는 가상화폐를 설명하기 위해 발표한 논문을 통해 많이 알려지게 되었다[1].

블록체인 기술은 PKI(PKI: Public Key Infrastructure) 구조를 이용하여 개인키, 공개키 쌍을 생성한 후 사용자에게 인증키로 부여한다. 개인키는 거래가 유효한지 검증 시 사용하는 중요한 요소이기 때문에 개인키에 대한 관리가 블록체인 기술에서 주요하게 다루어야 하는 분야이다.

블록체인 환경의 경우 개인키를 분실하였을 경우 본인 외에 이를 증명해 줄 제 3자가 존재하지 않기 때문에 본인의 개인키를 기억하는 것 외에는 복구할 수 있는 방법이 없다. 이에 대한 방법으로 블록체인 지갑[2, 3], 부분 정보를 이용한 개인키 복구[4], 블록체인 기반에서 생체정보와 OTP(One Time Password)를 이용한 개인키의 복구[5], 블록체인에서 개인키 암호화 및 복구[6] 등이 대표적으로 연구되고 있다.

블록체인의 종류로는 퍼블릭, 프라이빗, 하이브리드 블록체인이 있다. 퍼블릭 블록체인은 확장성 제한, 제한된 프라이버시, 계약 검증 취약, 저장 제약, 지속적이지 않은 합의 메커니즘, 거버넌스 및 표준 취약, 부적절한 툴, 양자 컴퓨터의 위협 등 8가지 문제점이 제시되고 있다[7]. 따라서, 본 논문에서는 백업된 개인키를 보관하는 장소(Repository)는 미리 정해진 조직이나 개인들 혹은 운영자의 승인을 받은 조직이나 개인만이 참여할 수 있는 프라이빗 블록체인 환경으로 제한한다.

본 논문은 II장에서는 관련연구를 기술하고, III장에서는 개인키 백업 및 복구에 대한 알고리즘을 이용하여 설명하고, 백업 및 복구 과정을 정의한다. IV장에서는 본 논문에서 제안하는 방법에 대해 보안성을 분석하고, V장에서는 결론 및 추후 연구계획에 대해서 기술한다.

II. Preliminaries

1. Related works

개인키 및 주요 내용의 백업과 관련된 연구는 다양하지만 본 논문에서는 개인키의 백업 및 복원과 관련된 연구를 살펴보고 문제점을 기술한다.

1.1 Blockchain Wallet

Seong 등은 개인이 소유하고 있는 장치의 MAC 주소를 활용한 개인키 생성 및 복구 방안을 제안하였다[2].

개인 소유 기기의 MAC 주소 및 사용자 본인이 기억하기 쉬운 4자리의 암호를 활용하여 개인키 생성 및 복구 방안을 제안하였으며, 사용자는 지갑을 활용하여 개인키를 생성할 때 본인이 쉽게 활용할 수 있는 기기들의 MAC 주소와 비밀번호 4자리를 설정하여 개인키를 생성한다. 사용자가 개인키를 분실하였을 경우, 개인키 생성할 때 사용하였던 기기들의 MAC 주소와 비밀번호 4자리를 사용하여 개인키 복구를 진행한다. 그러나 사용자가 비밀번호를 잊었을 경우 이를 알려주는 방법이 없고, MAC 주소는 생체정보 등 다른 정보에 비해 도용이 쉽다.

1.2 Private Key Recovery Scheme Using Partial Knowledge

Singh 등은 개인적인 지식 사용을 통해 암호화된 개인키를 복구하는 방법을 제안하였다[4].

Shamir's Secrete Sharing(SSS) 스킴[13]에 따라 개인이 원본 질문의 일부분만이라도 정확히 대답을 할 수 있다면 개인키 원본을 복구할 수 있고, 모든 요구된 정보는 보호된 개인키 자신 내에 저장되어 있기 때문에 PKRS(Partial Knowledge Recovery Scheme)는 복구 절차를 위해 어떠한 외부 서비스를 요구하지 않는다고 제안하였다. 그러나 개인키 소유자가 사망하거나 사고나 질병으로 인해 정보를 기억 할 수 없다면 제안하는 방법을 사용할 수 가 없다.

1.3 Private Key Encryption and Recovery in Blockchain

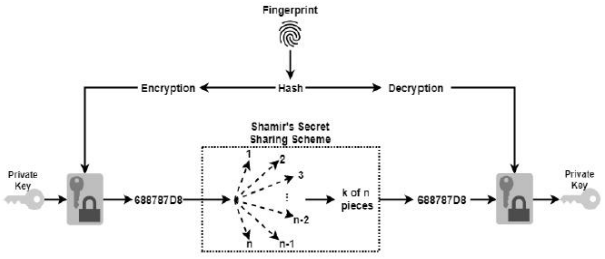

Fig. 1. Key Recovery

Aydar 등은 Fig. 1과 같이 소유자의 생체정보를 개인키와 같이 암호화하고 생체정보는 지문을 Reed-Solomon error correction[8] 방법을 이용하여 Encoding과 Decoding을 하였다[6].

개인키 복구는 Shamir's Secrete Sharing(SSS) 스킴[11]의 성질을 이용하여 복구하는 과정을 거친다. 본 논문에서 제안하는 방법과 일부 유사하나 본 논문은 Private

Blockchain 방법과 생체정보를 이용한 개인키 복원방법을 이용하여 보다 안전한 개인키 저장 및 복구 방법을 제시한다.

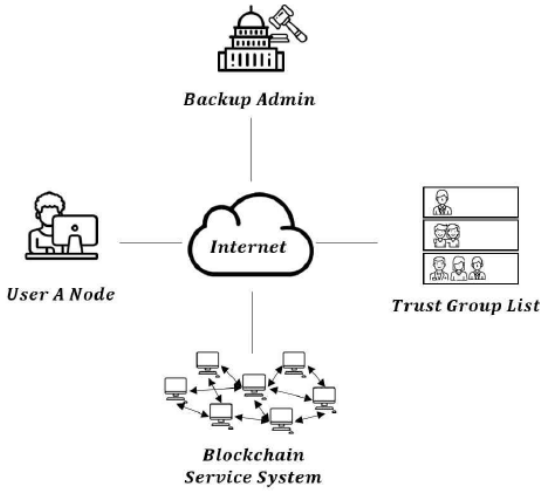

Fig. 2. Participants in private key backup framework

Yoon 등은 Fig. 2와 같이 자신의 개인키를 블록체인 시스템에 백업하려는 사용자로 개인키 백업을 실행하는 시스템을 제안하였다[9]. 사용자 A는 자신의 노드에서 개인키의 백업을 위해 신뢰 그룹 목록과 백업 관리자를 지정하고, 이후 사용자 A의 개인키는 설정에 따라 복제되고 분할되어 블록체인 시스템에 분산 저장된다. 백업 관리자는 복원 요청자의 복원 권한 및 기타 정책적인 검증을 수행한 후 개인키 복원에 필요한 정보를 제공한다.

그러나 생체정보 사용시 블록체인의 특징인 익명성을 제한한다고 주장하였지만 생체정보를 암호화하여 블록체인에 저장하기 때문에 익명성에 대한 제한은 없다. 또한 생체정보 수집에 높은 비용 또한 문제점으로 제안하였지만 이 또한 스마트폰에 장착된 센서를 통해 낮은 비용으로 수집이 가능하다.

다른 사용자(Backup Admin(BA))에게 자신의 개인키 복원 권한을 위임하는 것은 BA에 대한 신뢰성이 보장되지 않는다면 위험한 방법이 될 것이다.

1.4 etc

Zhu가 제안한 HA-eWallet은 블록체인 네트워크 밖에 다른 신뢰 기관을 두고, 개인키 생성 및 복구를 진행하는 연구를 진행하였다[10]. 네트워크 외부에 재해복구 센터를 두어서 개인키 생성 및 복구를 진행하였다. 해당 논문에서는 개인키를 생성할 당시에 여러 개의 개인키를 생성하고, 생성된 다수의 개인키는 재해 복구 센터에 저장한다. 개인키를 사용할 때는 재해복구 센터에 기록된 개인키 중 한 개만 사용하면 되는 방식이며, 개인키를 복구 할 때는 재해복구 센터에서 본인 인증을 받아 개인키를 복구한다.

OTP 기반 계정 복구 문자열 관리 체계는 모바일 기기의 유심 식별 번호 및 전화번호를 추출하여 개인키를 생성한다[3]. 개인키 복구를 위해서는 스마트 컨트랙트를 활용하여 OTP를 발급하고, 해당 발급된 OTP와 사용자의 핸드폰을 통해서 블록체인 네트워크와 IPFS(Inter Planetary File System)을 활용하여 개인키를 복구하는 방안을 제안하였다.

III. The Proposed Scheme

Fig. 3은 본 논문에서 제안하는 방법의 그림이다. 사용자는 생체정보를 센서를 통해서 단말기에 인식시킨다. 단말기의 센서를 통해 획득한 생체정보는 사용자의 개인키와 함께 폐쇄된 프라이빗 블록체인 내에 위치한 임의의 저장소에 백업된다. 여기서 임의의 저장소는 Fig. 4와 같이 독립된 블록체인으로 구성되어 있다.

생체정보를 이용한 개인키 백업의 방법에는 다음 세가지 방법이 있다[6].

- 개인키를 사용하기 위해 생체인식 인증을 사용
- 저장된 개인키는 생체인식 데이터와 함께 항상 암호화
- 개인키는 생체인식 데이터를 DES, RSA와 같이 알려진 암호화 알고리즘에 구현하여 생성

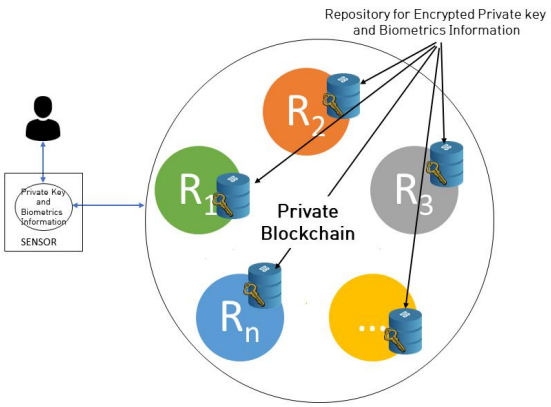

Fig. 3. Overview of proposed scheme

본 논문에서는 위 세 가지 방법을 사용하고 개인키 복구를 위해 생체정보를 이용한다. 사용자와 블록체인 내의 저장소 간에 대칭키 전달은 비대칭키를 사용하고, 이를 통해 공유된 대칭키는 사용자의 개인키와 생체정보를 암호화하

고 복호화하여 백업 및 복원에 사용된다.

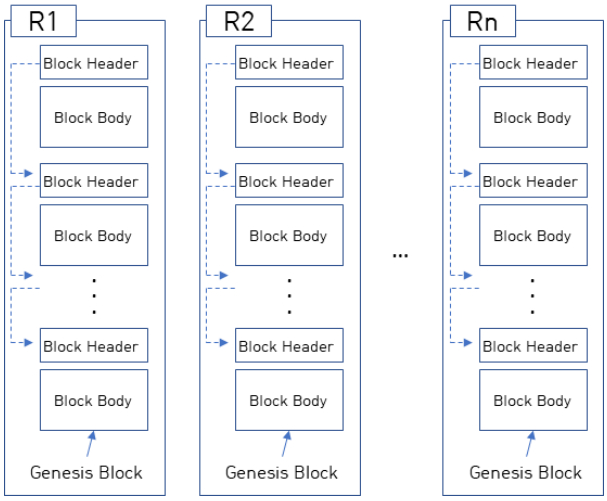

Fig. 4. Repository for backing up private keys based on Private Blockchain

생체인식 서명으로 생체정보를 사용하는 경우 생체정보는 신체의 일부이기 때문에 도용을 당한다면 상당히 위험한 상태가 된다. 그러므로 본 논문에서는 취소가능한(cancellable) 생체정보 시스템을 만들기 위해 단방향이면서, 되돌릴 수 없는(irreversible) 함수를 사용하여 특징(minutuate)을 변환하기 위해 카르테시안 변환(cartesian transformatuon)을 적용한다[6]. 사용자의 장치와 블록체인 내의 저장소에는 이미지 원본을 저장하지 않고, 변환 파라미터에 의해 변환된 내용을 저장한다.

본 논문에서 제안하는 수식을 간단하고 명확하게 하기 위해 Table 1.과 같이 기호를 정의한다.

Table 1. Notations

Notation	Description
$randint(x,y)$	Generate random numbers from min x to max y
RN	A large enough random number
BI_A	Biometric Information of User A
BKI_A	Concatenate user A's private key and biometric information
PR_A	Private Key of User A
PU_A	Public Key of User A
PR_{R_x}	Private Key of R_x
PU_{R_x}	Public Key of User R_x
$split(X,Y)$	split X by Y
$R[i]$	i-th repository on the Blockchain
$ReqRepsPU(X)$	Request the public key of the X-repository on the Blockchain
$CreateSK()$	Generate Secret Key
$x \overset{\subset}{\ll} n$ or $x \overset{\supset}{\gg} n$	Circular shift left x by n or Circular shift right x by n
$SndReps_x(Y)$	Send y to a specific repository x on the Blockchain
$A \overset{?}{=} B$	Compare A with B
$A \| B$	Concatenate A and B
$Hash(X)$	Hash X using a strong one-way hash function
$RcvU_X(Y)$	Receive Y from User X
A'	Hash value of A
$SK_{E_{XY}}(A)$	Encrypt A with the symmetric key shared by X and Y
$SK_{D_{XY}}(A)$	Decrypt A with the symmetric key shared by X and Y
$AdBC(X)$	Add X to Blockchain
$IsValid(X)$	Check if X is valid
$ExtrBC(X)$	Extract X from a specific storage in the blockchain
A_{PR}	The restored private key of A
$TMPA_{PR}$	Temporary private key of restored A

1.1 Blockchain Wallet

Shamir's Secrete Sharing(SSS) 스킴에 따르면, 데이터 D는 n개의 조각으로 나뉘고, D의 k개 조각은 D로 재조립이 가능하지만 $k-1$개의 조각으로는 D에 대한 정보를 알 수 없다[11].

Shamir's $(k,\ n)$ 임계치 스킴에 따라 데이터 D는 $D_1 \cdots D_n$개의 조각으로 나누고, D의 k 조각은 D로 재구성된다.

SSS의 스킴을 정리하면 다음과 같다.

- 사용자가 임의의 k 혹은 D_i 개 이상을 알고 있다면 쉽게 데이터 D를 계산할 수 있다.
- 사용자가 임의의 $k-1$ 혹은 D_i개 보다 적게 알고 있다면 데이터 D를 완전히 결정할 수 없다.

효율적인 임계치 스킴은 암호화 키 관리에 매우 유용하기 때문에 n = $2k$ - 1인 $(k,\ n)$ 임계치 스킴을 사용하면 매우 강력한 키 관리 스킴을 만들 수 있다.

n개 중 $\lfloor n/2 \rfloor = k-1$가 손실되어도 원본 키를 복구할 수 있지만 남아 있는 k개 중 $\lfloor n/2 \rfloor = k-1$가 노출될지라도 키를 재제작할 수 없다.

블록체인 내의 독립된 저장소를 $R(R_1,\ R_2,\ \ldots\ R_n)$이라고 하고 각 저장소에 키를 분산 백업을 아래와 같이 한다면 SSS의 성질에 따라 특정한 저장소의 정보가 노출이 되어도 안전하다.

$$R_1 \le k-1,\ R_2 \le k-1,\ \ldots\ ,\ R_n \le k-1 \quad (1)$$

$$k \le R_1 + R_2 \le n \quad (2)$$

$$k \le R_1 + R_2 + \ldots + R_n \le n \quad (3)$$

식 (1)은 각 저장소(R)에 $k-1$개의 사용자의 분할된 D가 저장되는 것을 의미하고, 식 (2)는 최소 2개 이상의 저장소의 정보를 이용해야 사용자의 D를 복구할 수 있다. 식 (3)은 모든 저장소의 정보를 모아도 k는 임계치 n보다 작거나 같다.

다음은 사용자의 장치에서 사용자의 개인키, 생체정보와 저장소의 위치 및 공개키를 이용하여 백업하는 과정이다.

Table 2. Creative - user

Algorithm 1: Creative - user
Input: user's private key, user's biometric information of user, Repository location and public key
Output: $k-1$ number of partitioned and encrypted private keys and biometric information for each repository, Symmetric key between user and repository
1: $RN = randint(2, R_{mx})$
2: $BKI_A = BI_A \| PR_A$
3: $B_A = split(BKI_A, RN)$
4: **for** i = 0 **to** $RN-1$ **do**
5: $R[i] = ranint(1, R_{mx})$
6: **end for**
7: **for** i = 0 **to** $RN-2$ **do**
8: $PU_{R[i]} = ReqRepsPU(R[i])$
9: $SK[R[i]] = CreateSK()$
10: $S[R[i]] = B_A \overset{\subset}{\ll} 1$
11: $S[R[i]]' = Hash(S[R[i]])$
12: $SndReps_{R[i]}(PU_{R[i]}(SK_{A,R[i]}[R[i]]))$
13: $SndReps_{R[i]}(SK_{E_{A,R[i]}}(S[R[i]], PR_A(S[R[i]]'), PU_A))$
14: **end for**
15: $PU_{R[RN-1]} = ReqRepsPU(S[R[RN-1]])$
16: $SK[R[RN-1]] = CreateSK()$
17: $S[R[RN-1]] = B_A \overset{\subset}{\ll} 1$
18: $S[R[RN-1]]' = Hash(S[R[RN-1]])$
19: $SndReps_{R[RN-1]}(PU_{R[RN-1]}(SK[R[RN-1]]))$
2 0 :
$SndReps_{R[i]}(SK_{E_{A,R[RN-1]}}(S[R[RN-1]], PR_A(S[R[RN-1]]'), PU_A));$

알고리즘 1은 다음과 같이 동작한다.

우선 사용자의 정보(개인키와 생체정보)를 저장할 저장소(Repository)의 개수와 위치를 정한 후, 정보를 저장소의 개수만큼 분할한다. 예를 들어, 사용자 A가 자신의 개인키(PR_A)와 생체정보(BI_A)를 5개의 저장소에 저장하고자 한다면 다음과 같다.

A는 자신의 개인키와 생체정보를 k로 나눈다. 이를 PR_{A_k}라 하면 $k \leq R_n$을 만족해야 한다. 여기서 R은 저장소이고 R_n은 n개의 저장소임을 의미한다.

A가 분할된 정보를 5군데 저장한다고 가정하면

$$PR_A = PR_{A_1} \| PR_{A_2} \| PR_{A_3} \| PR_{A_4} \| PR_{A_5} \quad (4)$$

$$R_1 = PR_{A_1} \| PR_{A_2} \| PR_{A_3} \| PR_{A_4} \quad (5)$$

$$R_2 = PR_{A_2} \| PR_{A_3} \| PR_{A_4} \| PR_{A_5} \quad (6)$$

$$R_3 = PR_{A_3} \| PR_{A_4} \| PR_{A_5} \| PR_{A_1} \quad (7)$$

$$R_4 = PR_{A_4} \| PR_{A_5} \| PR_{A_1} \| PR_{A_2} \quad (8)$$

$$R_5 = PR_{A_5} \| PR_{A_1} \| PR_{A_2} \| PR_{A_3} \quad (9)$$

여기서 각 저장된 R_n에는 $k-1$개의 분할된 정보가 저장된다. SSS에 의해 각 저장소에 있는 $k-1$개의 정보를 안다고 해도 복원할 수 없다.

Table 3. Save - Repository

Algorithm 2: Save - Repository
Input: Symmetric key between user and repository, user's private and public key, repository location
Output: encrypted repository location and Stored user's encrypted private key and biometric information
1: $RcvU_A(PU_{R[i]}(SK[R[i]]))$
2: $RcvU_A(SK_{E_{A,R[i]}}(S[R[i]], PR_A(S[R[i]]'), PU_A)))$
3: $PR_{R[i]}(SK[R[i]])$
4: $SK_{D_{A,R[i]}}(S[R[i]], PR_A(S'[R[i]]), PU_A)$
5: $PU_A(PR_A(S[R[i]]')$
6: **if** $Hash(S[R[i]]) \overset{?}{=} S[R[i]]'$ **then**
7: $AdBC(SK[R[i]], S[R[i]])$
8: **end if**

알고리즘 2는 다음과 같이 동작한다.

저장소는 사용자로부터 사용자와 저장소간의 대칭키, 사용자의 개인키, 공개키, 저장소 위치를 전송받는다. 여기서 사용자의 공개키는 백업시에만 받아서 사용자의 개인키로 암호화된 해쉬값을 복호화하는데 사용한다.

각 저장소내의 블록체인에 사용자와 저장소간 공유하고 있는 대칭키, 사용자의 개인키와 생체정보를 분산 저장한다.

1.2 Restoration of private key using biometric information

개인키 제공자는 R_1 ... R_n 중 2개 이상의 저장소에 본인의 개인키 정보와 생체정보를 요청하여 복원한다.

Table 4. Restoration - user

Algorithm 3: Restoration - user
Input: encrypted repository location, stored user's biometric information and encrypt hashed biometric information with the public key of the repository
Output: restored user's private key and biometric information

1: **for** i = 0 **to** $RN-1$ **do**
2: $ReqRepos_{S[R[i]]}(SK_{E_{A,R[i]}}(PU_{R[i]}(BI_A'), BI_A))$
3: $Rcv_{R[i]}(E_{SK_{A,R[i]}}(PR_{R[i]}(S[R[i]], BI_A'), BI_A))$
4: **if** $Hash(BI_A) \stackrel{?}{=} BI_A'$ **then**
5: $RS[i] = S[R[i]] - BI_A$
6: **end if**
7: **end for**
8: **for** i = 0 **to** $RN-1$ **do**
9: **if** $IsValid(RS[i])$ **then**
10: **for** $j=0$ **to** $RN-2$ **do**
11: $TMPA_{PR}[i] = RS[i] + RS[i+1]$
12: **end for**
13: **end if**
14: **end for**
15: $A_{PR}[RN-1] = RS[RN-1] + RS[0]$
16: $k=1$
17: **for** $i=0$ **to** $RN-1$ **do**
18: **for** $j=k$ **to** $RN-1$ **do**
19: **if** $TMPA_{PR}[i] \stackrel{?}{=} TMPA_{PR}[k]$ **then**
20: $A_{PR}[i] = TMPA_{PR}[i]$
21: $l=l+1$
22: **end if**
23: **end for**
24: $k=k+1$
25: **end for**
26: **if** $l \geq 2$ **then**
27: success
28: **else**
29: *fail*
30: **end if**

사용자는 분산 저장된 저장소에 사용자와 각 저장소간의 대칭키를 이용하여 해쉬화된 생체정보를 전송하여 본인의 백업된 개인키를 요청한다.

각 저장소로부터 $k-1$개의 분할된 정보를 수신한 사용자는 이를 복원한 후 유효한지 검사를 한다. 최종적으로 복원된 개인키가 2개 이상이 동일하다면 복구에 성공이고, 그렇지 않다면 실패다.

Table 5. Restoration - repository

Algorithm 4: Restoration - repository
Input: user's biometric information BI_A and hashed BI_A
Output: user's encrypted private key and biometric information

$RcvU_A(SK_{E_{A,R[i]}}(PU_{R[i]}(BI_A'), BI_A))$
$ExrtBC(SK[R[i]])$
$SK_{D_{A,R[i]}}(PU_{R[i]}(BI_A'), BI_A)$
if $PR_{R[i]}(PU_{R[i]}(BI_A')) \stackrel{?}{=} BI_A'$ **then**
$ExtrBC(S[R[i]])$
$SendU_A(SK_{E_{A,R[i]}}(PR_{R[i]}(S[R[i]], BI_A'), BI_A))$
end if

사용자로부터 사용자와 각 저장소간 공유된 대칭키로 암호화된 사용자의 생체정보와 함께 개인키 복구 요청을 받은 후 각 저장소는 저장소 내의 블록체인에서 대칭키를 추출하여 복호화한다. 각 저장소는 자신의 개인키로 사용자 정보를 암호화하고, 이를 대칭키로 암호화하여 사용자에게 전송한다.

IV. Security Analysis

보안성 분석에서는 사용자와 블록체인 간에 주고 받는 프로토콜, 사용자 장치, 저장소의 블록체인에 대해 안전하다는 것을 보인다.

본 논문에서 제안하는 방법에 대한 안전성 검증은 알고리즘 별로 각 단계에서 사용자와 각 저장소에서 주고 받는 메시지의 견고성으로 한다.

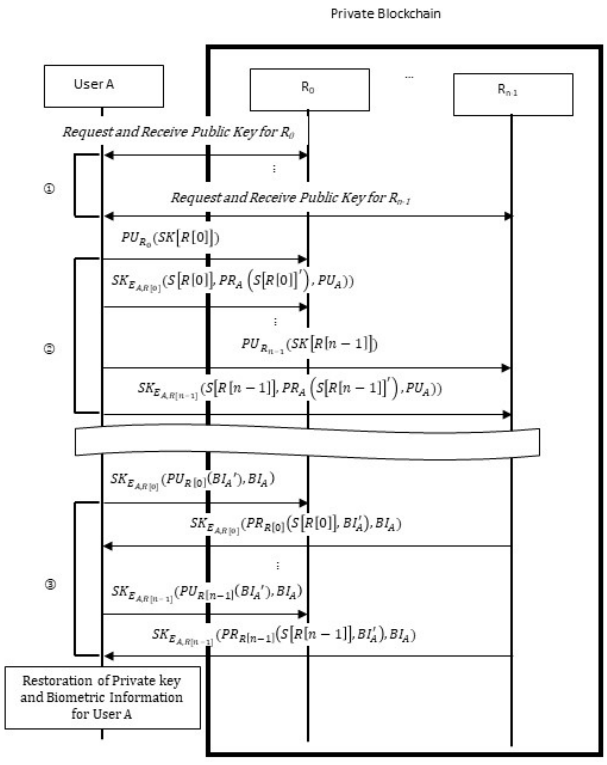

Fig. 5. safety verification

Fig 5.에서 ①은 사용자가 알고리즘 1에서 지정한 저장소에 각 저장소의 공개키를 요청하고 응답받는 것이다. 사용자는 모든 저장소로부터 공개키를 받는 것이 아니라 난수로 발생시킨 특정한 저장소로부터 받기 때문에 공격자로부터 안전하다.

②는 사용자가 수신한 특정 저장소들의 공개키로 사용자와 특정 저장소간의 공유할 대칭키를 전송하는 과정이다. 공격자는 저장소들의 개인키를 알지 못하므로 이를 복호화할 수 없기 때문에 중간자 공격으로부터 안전하다. 또한 사용자는 사용자와 특정 저장소간 공유하는 대칭키를 이용하여 분할된 $k-1$개의 사용자의 개인키, 생체정보와 이를 해쉬화한 후 사용자의 개인키로 암호화해서 전송한다. 이때 사용자의 공개키도 함께 저장소로 전송한다.

③은 사용자가 각 저장소로 자신의 생체정보와 해쉬화된 생체정보를 각 저장소의 공개키로 암호화한 후 이를 사용자와 각 저장소간 공유된 대칭키로 암호화하여 복구를 요청하는 것이다. 각 저장소는 사용자가 전송한 생체정보와 저장된 생체정보와 비교한 후 일치한다면 저장소 자신의 개인키로 사용자의 개인키와 해쉬화된 생체정보를 암호화하고 사용자의 생체정보와 함께 사용자와 각 저장소간 공유된 대칭키로 암호화하여 전송한다. 사용자는 수신한 데이터가 유효한지 검증한 후 유효하다면 개인키를 복구한다.

다음은 본 논문에서 제시하는 방법에 대해 개인정보 보호, 재생 공격, 중간자 공격, 데이터의 위·변조에 대해 안전성을 입증한다.

1.1 Information Security

본 논문에서는 사용자의 개인정보(생체정보)를 원본 데이터 자체를 사용하는 것이 아니라 [6]에서 제안한 취소가능한(cancellable) 생체정보 시스템인 단방향, 되돌릴 수 없는(irreversible) 함수를 사용하여 특징 (minutuate)을 변환하기 위해 카르테시안 변환 (cartesian transformatuon)을 적용하기 때문에 사용자의 장치와 블록체인 내의 저장소에는 이미지 원본을 저장하지 않고, 변환 파라미터에 의해 변환된 내용을 저장한다.

또한, 퍼블릭 블록체인이 아니라 프라이빗 블록체인에 암호화되어 저장되어 있기 때문에 일반인의 접근이 힘들 뿐 아니라 해킹이 되더라도 블록체인에 분산 저장되어 있는 모든 원장을 수정하여야 하고, 하나의 저장소 원장을 수정한다 하더라도 임계치(본 논문에서는 2개이며 조정이 가능함) 값 이상으로 수정하여야 하기 때문에 사실상 불가능하다.

또한 Fig. 5 어디에도 사용자의 개인키 혹은 생체정보 원본 데이터를 전송하지 않기 때문에 전송도중 메시지가 유출되어도 파급효과는 미비하다. 중간에 생체정보가 유출이 되더라도 변환된 템플릿이므로 보안에 안전하다.

1.2 Replay Attack

공격자가 사용자 장치와 블록체인 사이에서 송수신되는 ①~③에서 메시지를 중간에 가로채더라도 사용자 장치, 사용자 장치와 연결된 특정 블록체인의 개인키, 사용자와 저장소간 공유된 대칭키를 알지 못하므로 재생공격이 불가능하다.

또한 인증 수단인 생체정보는 OTT를 사용하고 있기 때문에 재사용이 불가능하다.

1.3 Man-in-the-Middle Attack

본 논문에서 제안하는 방식은 공격자가 사용자 장치와 임의의 블록체인 사이의 정보를 가로채더라도 사용자 인증 단계에서 해당 블록체인의 개인키를 알아야 하고 해당 블록체인의 개인키가 노출이 되어도 Fig. 4와 같이 블록체인에 분산되어 있는 모든 원장과 비교하여야 하기 때문에 중간자 공격은 불가능하다. 또한 모든 메시지는 암호화와 해쉬화되어서 전송되기 때문에 중간자 공격으로부터 안전하다.

1.4 Tamper Attack

사용자의 원본 생체정보(BI_A)는 각 단계에서 사용자 단말기에서 바로 삭제하기 때문에 공격자는 정보를 위·변조하더라도 개인키를 복구할 수 없다. Fig. 5의 각 단계를 공격하기 위해서는 공격자는 PR_A, PU_A, SK, PR_{R_x}, PU_{R_x}를 알아야 한다. 따라서, 공격자가 전송 중인 메시지를 중간에 가로챈다 하더라도 각 단계의 메시지를 위·변조할 수 없다.

V. Conclusions

본 논문에서는 Shamir's Secrete Sharing(SSS)의 스킴을 사용하여 안전하게 사용자의 개인키 백업 방법을 제안하였다. 본 논문에서는 사용자의 생체정보를 사용하여 OTT를 생성하고 이를 이용하여 블록체인에 백업된 개인키를 복구한다. 공격자가 사용자 장치를 불법으로 탈취하여도 저장된 생체정보가 없고, 블록체인을 해킹을 한다 하더라도 분산된 모든 원장의 OTT를 수정하여야 하기 때문에 사실상 불가능하다. 또한 중간에 메시지를 탈취하거나 전송 데이터를 위변조하기 위해서는 사용자, 저장소의 개인키와 공개키를 모두 알아야 하고, 사용자와 저장소간의 공유된 대칭키 역시 알아야 한다.

본 논문에서 제시하는 방법은 보안성 분석을 통해 보안 특성이 우수하고, 다양한 공격에 대해 안전함을 보였다.

또한 공격자가 한 개의 저장소에서 원본 데이터를 확보해도 임계치(본 논문에서는 2) 이상의 데이터를 확보해야 복원이 가능하기 때문에 임계치 미만의 데이터로는 개인키 복원은 원천적으로 불가능하다.

추후 연구과제로는 퍼블릭형과 하이브리드 형태의 블록체인에서 민감한 개인정보를 보다 안전하게 보호하고 저장할 수 있는 방법과 본 논문에서는 임계치를 2로 제안했지만 사용자의 환경에 따라 최적의 임계치를 구하는 내용에 대해 연구할 계획이다.

ACKNOWLEDGEMENT

This work has been supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT)(No. 2021R1F1A1048973).

REFERENCES

[1] Satoshi Nakamoto, “Bitcoin: A Peer-to-Peer Electronic Cash System,” Bitcoin.org, 2009. DOI: https://bitcoin.org/bitcoin.pdf

[2] Si Woo SEONG, Jung Won SEO, and Soo Yong PARK, “Private Key Generation and Recovery System Based on Personal Devices for Blockchain Wallet,” Proceeding of Korea Computer Congress 2022 (KCC2022), Korean Institute of Information Scientists and Engineers, pp.1294~1296, June, 2022. DOI: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11113642

[3] S.Song, S.Kim, J.Shin, etc. “Recovery phrase management scheme for public blockchain wallets based on OTP”. The Journal of the Institue of Internet, Broadcasting and Communication (IIBC). Vol.20, No.1, pp.35-44, Feb.29 (in Korean). DOI: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09012747

[4] Har Preet Singh, Kyriakos Stefanidis, and Fabian Kirstein, “A Private Key Recovery scheme Using Partial Knowledge,” 2021 11th IFIP International Conference on New Technologies, Mobility and Security (NTMS), Paris, France, 19-21 April, 2021. DOI: 10.1109/NTMS49979.2021.9432642

[5] Hyung-Jin Mun, “Biometric Information and OTP based on Authentication Mechanism using Blockchain,” Journal of Convergence for Information Technology Vol. 8. No. 3, pp. 85-90, 2018. DOI: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE09012747

[6] Mehmet Aydar, Salih Cemil C¸etin, Serkan Ayvaz, Betül Aygün, “Private Key Encryption and Recovery in Blockchain,” Elsevier, October 23, 2021. DOI: https://doi.org/10.48550/arXiv.1907.04156

[7] Da Hongfei, Blockchain and Smart Economic, Blockchain Partners Summit 2018, July, 2018.

[8] I. S. Reed and G. Solomon, “Polynomial codes over certain finite fields,” Journal of the society for industrial and applied mathematics, vol. 8, no. 2, pp.300~pp304, June, 1960. DOI: https://epubs.siam.org/doi/10.1137/0108018

[9] Ta-Yeon Yoon and Jong-Sub Moon, “Private Key Backup and Recovery Framework in Blockchain-based Service Environment,” Journal of Digital Contents Society, Vol. 20, No. 12, Dec., 2019. DOI: https://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE11113642

[10] F.Zhu, W.Chen, Y.Wang, etc. “Trust your wallet: a new online wallet architecture for bitcoin,”. 2017 International Conference on Progress in Informatics and Computing (PIC), pp.307-311, 2017. DOI: 10.1109/PIC.2017.8359562

[11] A. Shamir, “How to share a secret,” Communications of the ACM, vol. 22, no. 11, Nov., 1979. DOI: https://doi.org/10.1145/359168.359176

Authors

Seungjin Han received the B.S., M.S. and Ph.D. degrees in Computer Science and Engineering from Inha University, Korea, in 1989, 1992 and 2004, respectively. Dr. Han joined the faculty of the Department of e-Business at Kyung-In Women’s University, Incheon, Korea, in 2004. He is currently a Associate Professor in the Department of Software Convergence, Kyung-In Women’s University. He worked for Research Center of Daewoo Telecommunication as a TDX software developer from Jan. 1992 to Jun. 1996, and National Information Society Agency(NIA) as project manager from Jun. 1996 to Jul. and SKTelecom as project manager from Jul. 1996 to Jan. 1998. He was a lecturer of Inha University from Mar. 2002 to Feb. 2004. He is interested in computer network and security. His research has always been in the area of MANET and Sensor Networks or technologies which relate to it, such as Security, Protocol and Routing in MANET and USN, Middleware in USN and Security using Biometric Systems.