JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 22, No. 8, pp. 1281-1288, Aug. 2021

Check for updates

블록체인 기반의 E-Health 기록 및 공유 스킴의 취약성 분석

김 종 건[1] · 최 윤 성[2*]
[1]인제대학교 컴퓨터공학부 학사과정
[2*]인제대학교 AI융합대학 조교수

Weakness of blockchain-based electronic-health recording and sharing scheme

Jong-Geon Kim[1] · Youn-Sung Choi[2*]

[1]Undergraduate Course, Department of Computer Science, Inje University, Gimhae-si 50834, Korea
[2*]Assistant Professor, AI Convergence College, Inje University, Gimhae-si 50834, Korea

[요 약]

웨어러블 및 모바일네트워크의 발전으로 원격 의료서비스 등이 급격히 늘어남에 따라 환자의 E-Health Redord(EHR)를 효율적이고 안전하게 관리하는 시스템에 대한 관심이 늘어나게 되었다. 의료정보를 중앙집중식 저장방식으로 처리할 때 발생하는 다양한 문제를 해결하기 위해 블록체인 기술을 활용하여 의료정보를 저장 및 공유하는 시스템에 대한 연구가 본격화되었고, Shamshad 등은 컨소시엄과 프라이빗 방식을 함께 사용하는 블록체인기반 의료정보 저장 및 공유 스킴을 제안했다. 본 논문에서는 Shamshad등이 제한한 스킴의 동작과정 및 취약점을 취약점을 분석하여, Shamshad 등이 제안한 스킴이 오프라인 패스워드 추측공격, 내부자공격, 완전 순방향 비밀성 미충족, 공격자에 의한 사용자 로그인제한 가능, 동작과정의 비트수 불일치 등의 취약점이 발생할 수 있는 것을 분석했다.

[Abstract]

With the rapid increase of telemedicine services due to the development of wearables and mobile networks, interest in a system that efficiently and safely manages a patient's E-Health Redord (EHR) has increased. However, due to the characteristics of medical information, centralization was not suitable, so research on a blockchain-based medical information storage and sharing system began. Shamshad et al. proposed a blockchain-based medical information storage and sharing scheme that uses a consortium and a private. In this paper, as a result of analyzing the schemes of Shamshad et al., it was found that there are vulnerabilities in offline password guessing attack, insider attack, complete forward secrecy not satisfied, user login restriction by attacker is possible, and bit number mismatch in operation process.

색인어 : 안전성 분석, 세션 개시 프로토콜, 사용자 인증 스킴, 블록체인

Key word : Security analysis, Session initiation protocol, User authentication scheme, Blockchain

http://dx.doi.org/10.9728/dcs.2021.22.8.1281

Received 13 July 2021; **Revised** 23 August 2021
Accepted 23 August 2021

***Corresponding Author; Youn-Sung Choi**

Tel: +82-054-320-3206
E-mail: cys2020@inje.ac.kr

Ⅰ. 서 론

정보 통신 기술의 급격한 발전으로 Telecare Medicine Information System(TMIS)를 통해 환자에게 의료 서비스를 제공하게 되었다. TMIS를 통해 의사와 환자가 직접 만나지 않고도 환자의 증상에 대해 논의하고 다른 의료전문가와 중요한 정보를 교환 할 수 있어 환자에게 큰 편의를 제공하며, 치료비용을 크게 줄인다. 또한 E-Health 시스템에서 환자의 가장 최근 건강상태에 대한 정보는 의사의 의료행위 결정을 효율적으로 도와줄 수 있다. 그러나 새로운 환자의 경우 해당 의사는 환자의 이전 병력 및 기타 관련 정보에 실시간으로 접근할 수 없기 때문에 의료진이 정확하게 의학적 진단과 치료를 하는 것은 어려울 수 있다. 또한 E-Health Record(EHR)에서 데이터 무결성, 개인정보 보호 및 기밀성을 보장하는 것은 매우 중요하다. 그러나 이러한 EHR 시스템의 요구사항은 시스템이 호스팅 되는 관할 지역의 개인정보 보호법에 따라 달리질 수 있다[1].

의료정보를 처리하기 위한 필요하다고 판단되는 다양한 요구사항을 충족하기 위해서 많은 연구자들이 개인정보 보호 데이터 공유 계획을 제시했으나 기존의 데이터 공유방식은 주로 중앙 집중식 시스템에 의존해왔다. 기존의 중앙집중식 시스템은 환자의 의료기록을 의료기관에서 관리하고 있지만, 이를 규제하는 규제기관에서는 의료데이터에 직접 접근할 수 없는 구조를 갖추기고 있기 때문에 의료정보가 불법적으로 사용될 수 있다는 위험성을 내포하고 있었다. 따라서 중앙집중식 시스템을 기반으로 하는 개인의 민간한 정보라 할 수 있는 의료 데이터를 공유하는 방식은 적합하지 않다[2,3].

최근 블록체인 패러다임이 새로운 이슈로 떠오르면서 위에서 언급한 문제를 해결하기 위한 이해관계자들의 관심이 집중되고 있다. 블록체인의 분산 저장 시스템을 이용해 EHR의 효율적인 사용을 위한 플랫폼을 효과적으로 구현할 수 있으며 궁극적으로 의료정보를 비가역적이고 영구적으로 저장할 수 있다. 이러한 특성으로 인해 블록체인 기술을 통해 EHR 에 대한 효율적이고 강력한 공유 체계를 구현할 수 있다. 그리하여 블록체인 기술을 적용한 다양한 스킴이 제안되었고, Shamshad 등은 컨소시엄과 프라이빗 방식을 함께 사용하는 블록체인기반 의료정보 저장 및 공유 스킴을 제안했다[4-6].

본 논문에서는 그중 Shamshad 등이 제안한 스킴을 분석하고 그 과정에서 오프라인 패스워드 추측공격, 내부자 공격에 취약하고 완전 순방향 비밀성을 충족하지 않으며 공격자에 의해 정당한 사용자의 로그인이 제한될 수 있으며 동작과정에서 비트수가 불일치 할 수 있다는 것을 밝혀냈다.

본 논문의 구성은 먼저 2장에서 Shamshad 등이 제안한 스킴을 이해하기 위해 필요한 관련 연구들에 대해 설명하고 3장에서 Shamshad 등이 제안한 스킴의 등록 및 로그인 및 인증 과정을 분석하고 4장에서 Shamshad 등의 프로토콜에 대한 취약점 분석을 통해 밝혀진 문제점에 대해 설명한다. 그리고 마지막 5장에서 본 논문의 결론으로 논문을 마무리한다.

Ⅱ. 관련 연구

TMIS에서 환자의 익명성은 매우중요하며 개인정보의 침해는 매우 심각한 문제가 될 수 있다. 따라서 다양한 암호화 기술이 환자의 익명성 보장과 개인정보보호에 사용되었으나 기존의 2인 기반 프로토콜에 존재하는 여러 가지 문제로 인해 3인 기반 상호 인증 체계의 개념이 도입되었다. 3자 인증 체계는 부인, 추측, 중복 및 키 탈취 문제 해결할 수 있었으나 이러한 프로토콜 중에도 취약한부분이 많거나, 필수적인 기능이 결여되어 있는 경우가 많았고, 이를 통해 3자 기반 인증 프레임워크를 위한 효율적이고 안전한 정보보호 설계가 매우 어려운 일이라는 것을 알 수 있었다[7-15].

또한 PUF(Physical Uncloneable Function) 물리적 복제 방지 기술을 기반으로 하는 여러 프로토콜이 존재한다.이러한 PUF 기반 프로토콜은 여러 물리적 위협에 대해 안전하여 복제 장치를 계층적으로 보호할 수 있어 복제 장치를 도난당한 경우에도 공격자는 여전히 해당 복제 장치에서 PUF를 도출할 수 없다.[16] 그럼에도 PUF를 사용한 프로토콜은 다른 관련 장치가 취약하다면 여전히 안전하지 않다.

따라서 위에서 언급한 문제들을 해결하기 위해 최근에 급부상한 블록체인 기술을 EHR에 이용한 여러 가지 솔루션이 발표되었다. Chen 등이 블록체인을 기반으로 한 의료데이터 서비스 공유를 위해 제3의 신뢰할 수 없는 당사자에게 의존하지 않고 환자의 프라이버시 와 데이터의 안전한 저장을 달성할 수 있는 구현체계를 강조하는 프레임워크를 제안했다. 또한 Zhang 등은 환자의 EHR 데이터의 개인정보 보호 및 보안을 위한 블록체인 기반 프로토콜을 제안하고 명확한 체계와 시스템 모델을 도입했다. 이들의 제안은 컨소시엄 블록체인 또는 프라이빗 블록체인으로 구성되었다[17,18].

그리고 본 논문에서 다룰 Shamshad 등은 Zhang 등이 제안한 시스템 모델을 참조하여 설계모델에 프라이빗 및 컨소시엄 블록체인을 함께 사용하였고 또한 PUF 기술도 사용하였다.Shamshad 등이 설계한 스킴은 크게 세가지 디지털 서명 알고리즘과 퍼지추출 기술이 사용된다.

① KeyGen(Bi) :사용자의 생체인식 B를 받아 개인키와 공개키 쌍을 생성한다.
② sig(pri, msg) ー> s :이 함수는 개인키 pri 및 메시지msg 입력에 해당하는 디지털서명 S를 계산한다.
③ Ver(pub, S, msg) ->bϵ {1, 0}: 이 함수는 S값이 공개키 pub 및 메시지 msg에 해당하는 유효한 서명인지를 검증한다.
④ Fuzzy Extractor : 퍼지추출은 키 생성 및 재생성 프로세스에 따라 Gen, 및 Rep 함수 쌍으로 사용된다.Gen은 입력으로 제공된 생체정보 B에 대해 β, β^* 을 생성한다. 그리고 Rep 는 입력 생체정보 B와 β^* 를 통해 키 β 를 출력하는 함수 이다.

Ⅲ. Shamshad 등의 스킴 분석

이 장에서는 Shamshad 등 의 스킴의 등록 및 인증 동작과정을 분석한다. 동작과정에서 사용되는 용어들에 대한 설명은 그림1 과 같다.

표 1. 용어 설명
Table 1. Common used notations.

Notations	Elucidations
SM_j	System manager of infrastructure
sr	Secret key of system
U_{pi}	Patient of the system
ID_{pi}	Specific user's identity
B_{pi}	User's Bio-metric impression
$CRP(C_{pi},R_{pi})$	Challenge response pairs
U_{ui}	Data user of the system (Acting as third-party)
t_{pi-j}	Validity period for user
MD	Onboard memory device of each data user
MS	Medical server
$\mathcal{EVE}$ adv	The adversary
SID	SM's identity
Gen(), Rep()	Fuzzy extractor algorithms
pri_{pi}, pub_{pi}	Private and public key pairs of Upi
PW_{pi}	User's Password
BN_{pi}	Block number of user
PUF_{pi}	Physically uncloneable functions
h()	One-way digest function of hashing
||	Concatenation operator

3-1 사용자 등록 과정

등록과정은 각 사용자가 처음 서버에 등록할 때 한번 수행된다. 그림1은 등록과정을 간략하게 나타내며 자세한 등록과정은 다음과 같다.

① 사용자의 *ID, PW,* 그리고 생체정보 *Bpi*를 입력받고 임의의 숫자 n을 생성한다. 그리고 *Gen(Bpi)*를 사용해 생체정보를 암호화 하여 βpi와 $\beta^* pi$를 생성한다. 그리고 KeyGen(Bpi)로 개인키,공개키 를 아래와 같이 생성 및 계산한다.

$$MPW = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}$$
$$S_{pi} = sig(pri, ID \| REV \| n_{pi} \oplus \beta_{pi}) \quad (1)$$

그리고 보안채널을 통해 서버로 $\{ID, MPW, S, n_{pi1} \oplus \beta_{pi}, pub, pri\}$ 전송한다.

② 서버는 전달받은 Spi 인증서의 유효성을 검증 후 다음과 같이 $S = sig(sr, ID_{pi}, SID_j \| pub_{pi} \| Rev_{pi} = 0)$ 를 계산하고, 블록체인 네트워크에 $ID, pub_{pi}, SID_j, pub_j, Rev_{pi}, S_{pi}, S_{pi-j}$를 생성 하고 Broadcast 후 새로운 블록넘버 BN_{pi} 생성하고 다음과 같이 w, V_{pi-j}, U_{pi-j} 을 계산한다.

$$w = h(sr \| SID_j)$$
$$V_{pi-j} = h(ID \| t_{pi-j} \| w)$$
$$U_{pi-j} = V_{pi-j} \oplus MPW \quad (2)$$

그리고 DID_{pi}와 C_{pi} 를 생성한 후, $< = (ID_{pi} \oplus MPW_{pi}) \oplus (n_{pi1} \oplus \beta_{pi})$ 계산한 후, 보안채널을 통해 $\{DID_{pi}, U_{pi-j}, t_{pi-j}, <, C_{pi}, BN_{pi}\}$를 사용자에게 전달한다.

③ 사용자는 수신 받은 메시지를 바탕으로 다음의 계산과정을 수행하고 디바이스에 $\{DID_{pi}, W_{pi-j}, <, a, B^*_{pi}\}$ 저장한다.

$$Extract\ R_{pi} = PUF_{pi}(C_{pi})$$
$$\alpha = (ID_{pi} \| PW_{pi}) \oplus (n_{pi1} \| pri_{pi} \| t_{pi-j} \| BN_{pi}) \oplus \beta_{pi}$$
$$W_{pi-j} = U_{pi-j} \oplus n_{pi1} \quad (3)$$

④ 사용자는 보안채널을 통해 SM_j에게 R_{pi}를 전달하고 서버는 $\{ID, <pri_{pi}, pub_{pi}>, (C_{pi}, R_{pi})\}$를 DID를 식별자로 하고 서버키 sr 로 암호화 하여 저장한다.

3-2 로그인/인증 과정

로그인 과정은 등록된 사용자가 서버에 로그인하여 인증하고자 할 때 사용된다. 그림2는 로그인/인증 과정을 간략하게 나타내며 자세한 과정을 다음과 같다.

① 사용자의 ID와 PW 그리고 생체정보 B_{pi}를 입력, 다음을 계산 한다.

$$Rep(B_{pi}, \beta^*_{pi}) = \beta_{pi}$$
$$(n_{pi1} \| pri_{pi} \| t_{pi-j} \| BN_{pi}) = (ID_{pi} \| PW_{pi}) \oplus \alpha \oplus \beta_{pi} \quad (4)$$
$$MPW_{pi} = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}$$

그리고 $< =? (ID_{pi} \oplus MPW_{pi}) \oplus (n_{pi1} \oplus \beta_{pi})$를 통해 입력값이 유효한지 검증한다.

② 그 후 계산과정을 거친 후, 로그인요청 $\{DID_{pi}, D_1, D_2\}$를 서버에 전달한다.

$$U_{pi-j} = W_{pi-j} \oplus n_{pi1}\ V'_{pi-j} = U_{pi-j} \oplus MPW_{pi}$$
$$Generates\ random\ number\ npi2$$
$$computes: D_1 = (n_{pi2} \| t_{pi-j}) \oplus (ID_{pi})$$
$$D_2 = h(ID_{pi} \| n_{pi2} \| V'_{pi-j} \| BN_{pi} \| t_{pi-j}) \quad (5)$$

③ 사용자의 로그인 요청에 대한 서버는 DID와 sr을 사용해 데이터베이스에서 {IDpi,(Cpi, Rpi)} 가져오고, 다음을 수행한다.

U_{pi}	SM_j
Select ID_{pi}, PW_{pi}	
Generate arbitrary number n_{pi1}	
Imprints B_{pi}	
$GEN(B_{pi}) = (\beta_{pi}, \beta^*_{pi})$	
$KeyGen(B_{pi}) \rightarrow (pri_{pi}, pub_{pi})$	
$\overline{MPW}_{pi} = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}$	
$S_{pi} = sig(pri_{pi}, ID_{pi} \| Rev_{pi} \| n_{pi} \oplus \beta_{pi})$	
(Secure Channel) $\{ID_{pi}, \overline{MPW_{pi}}, S_{pi}, n_{pi} \oplus \beta_{pi}, pub_{pi}, pri_{pi}\}$ →	
	$Ver(pub_{pi}, S_{pi}, ID_{pi} \| Rev_{pi} n_{pi1} \oplus \beta_{pi}) = ?1$
	$S_{pi-j} = sig(sr, ID_{pi}, SID_j \| pub_{pi} \| Rev_{pi} = 0)$
	Broadcasts $ID_{pi}, pub_{pi}, SID_j, pub_{pi}, Rev_{pi}, S_{pi}, S_{pi-j}$ to blockchain network
	Generate new block BN_{pi}
	$\omega_j = h(sr \| SID_j)$ $V_{pi-j} = h(ID_{pi} \| t_{pi-j} \| \omega_j)$
	$U_{pi-j} = V_{pi-j} \oplus \overline{MPW}_{pi}$
	Select DID_{pi} and Generate C_{pi}
	$<= (ID_{pi} \oplus \overline{MPW}_{pi}) \oplus (n_{pi1} \oplus \beta_{pi})$
← (Secure Channel) $\{DID_{pi}, U_{pi-j}, <, t_{pi-j}, C_{pi}, BN_{pi}\}$	
Extracts $R_{pi} = PUF_{pi}(C_{pi})$	
$\alpha = (ID_{pi} \| PW_{pi}) \oplus (n_{pi1} \| pri_{pi} \| t_{pi-j} \| BN_{pi}) \oplus \beta_{pi}$	
$W_{pi-j} = U_{pi-j} \oplus n_{pi1}$	
stores $\{DID_{pi}, W_{pi-j}, <, \alpha, \beta^*_{pi}\}$ in MD_{pi}	
(Secure Channel) $\{R_{pi}\}$ →	
	Keeps $\{ID_{pi}, < pri_{pi}, pub_{pi} >, (C_{pi}, R_{pi}), \}$ against DID_{pi} in database using sr

그림 1. 등록과정
Fig. 1. User registration process

$$(n_{pi2} \| t_{pi-j}) = D1 \oplus ID_{pi}, Checks\ t_{pi-j} \text{ and } BN_{pi}$$
$$calculates: V_{pi-j} = h(ID_{pi} \| t_{pi-j} \| \omega_j) \quad (6)$$

④ $D_2 = ?\ h(ID_{pi} \| n_{pi2} \| V_{pi-j} \| BN_{pi} \| t_{pi-j})$ 를 통하여 $D2$값을 검증하여 유효한 사용자의 요청인지 확인한다. 그리고 새로운 동적 ID $newDID_{pi}$를 생성하고 database에 업데이트 후 다음 계산 과정을 수행한 후, 사용자에게 $\{D4, D5, R_{pi}\}$를 전달한다.

$$Generates\ random\ number: n_j$$
$$Computes: SK = h(ID_{pi} \| SID_j \| n_{pi2} \| n_j \| R_{pi})$$
$$D_4 = (ID_{pi} \| n_{pi2}) \oplus (n_j \| DID^{new}_{pi}) \quad (7)$$
$$D_5 = h(ID_{pi} \| SID_j \| SK \| V_{pi-j})$$

⑤ 서버에게 전달받은 $\{DID_{pi}, D_1, D_2\}$로 다음과 같은 계산을 통해 사용자는 세션키를 생성하고, 서버의 메시지를 인증한다.

$$Extracts: R_{pi} = PUF_{pi}(C_{pi})$$
$$(n_j \| DID^{new}_{pi}) = h(ID_{pi} \| n_{pi2}) \oplus D4$$
$$SK = h(ID_{pi} \| SID_j \| n_{pi2} \| n_j \| R_{pi}) \quad (8)$$
$$D_5 = ?h(ID_{pi} \| SID_j \| SK \| V'_{pi-j})$$

⑥ $(n_j \| DID^{new}_{pi})$로 MD_{pi} 업데이트하고 인증후 계산한 세션키 SK를 향후 통신에 이용한다.
Updates $(n_j \| DID^{new}_{pi})$ in his MD_{pi}

U_{pi}	SM_j
Insert ID_{pi}, PW_{pi}	
Imprints B_{pi}	
$Rep(B_{pi}, \beta^*_{pi}) = \beta_{pi}$	
$(n_{pi1} \| pri_{pi} \| t_{pi-j} \| BN_{pi}) = (ID_{pi} \| PW_{pi}) \oplus \alpha \oplus \beta_{pi}$	
$\overline{MPW}_{pi} = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}$	
Verifies: $<= ?(ID_{pi} \oplus \overline{MPW}_{pi}) \oplus (n_{pi1} \oplus \beta_{pi})$	
$U_{pi-j} = W_{pi-j} \oplus n_{pi1}$	
$V_{pi-j} = U_{pi-j} \oplus \overline{MPW}_{pi}$	
Generates random number n_{pi2}	
$D_1 = (n_{pi2} \| t_{pi-j}) \oplus (ID_{pi})$	
$D_2 = h(ID_{pi} \| n_{pi2} \| V_{pi-j} \| BN_{pi} \| t_{pi-j})$	
$\{DID_{pi}, D_1, D_2\}$ →	
	Retrieves $\{ID_{pi}, (C_{pi}, R_{pi})\}$ against DID_{pi} using sr
	$(n_{pi} \| t_{pi-j}) = D_1 \oplus ID_{pi}$
	Checks $t_{pi} - j$ and BN_{pi}
	Computes: $V_{pi-j} = h(ID_{pi} \| t_{pi-j} \| \omega_j)$
	Verifies $D_2 = ? h(ID_{pi} \| n_{pi2} \| V_{pi-j} \| BN_{pi} \| t_{pi-j})$
	Select DID^{new}_{pi} and updates in its database
	Generate random number n_j
	$SK = h(ID_{pi} \| SID_{pi} \| n_{pi2} \| n_j \| R_{pi})$
	$D_4 = (ID_{pi} \| n_{pi2}) \oplus (n_j \| DID^{new}_{pi})$
	$D_5 = h(ID_{pi} \| SID_j \| SK \| V_{pi-j})$
← $\{D_4, D_5, C_{pi}\}$	
Extracts: $R_{pi} = PUF_{pi}(C_{pi})$	
$(n_j \| DID^{new}_{pi}) = h(ID_{pi} \| n_{pi2}) \oplus D_4$	
$SK = h(ID_{pi} \| SID_j \| n_{pi2} \| n_j \| R_{pi})$	
$D_5 = ? h(ID_{pi} \| SID_j \| SK \| V_{pi-j})$	
$SK = h(ID_r \| Q_r \| Q_s \| M_r)$	
Updates $(n_j \| DID^{new}_{pi})$ in his MD_{pi}	

그림 2. 로그인/인증 과정
Fig. 2. Authentication and key agreement process

Ⅳ. Shamshad 등의 스킴의 취약성 분석

본 논문에서는 Shamshad 등의 스킴의 동작과정을 분석하여 오프라인 패스워드 추측 공격, 내부자공격, 완전 순방향 비밀성(Perfect Forward Secrecy) 미충족, 공격자에 의해 정상적인 사용자의 로그인 제한 가능, 동작과정 비트수 불일치의 문제점이 있다는 것을 밝혀냈다.

4-1 Off-line Password Guessing Attack

사용자의 모바일기기에는 $\{DID_{pi}, W_{pi-j}, <, a, B^*_{pi}\}$ 가 저장되어있고 공격자가 이를 물리적인분석방법을 통해 알아냈다고 가정하자. 등록과정에서

$$<= (ID_{pi} \oplus MPW_{pi}) \oplus (n_{pi1} \oplus \beta_{pi}) \quad (9)$$

위의 수식을 통해 다음을 도출해 낼 수 있다.

$$
\begin{aligned}
&MPW_{pi} = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi} \\
&<= (ID_{pi} \oplus PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}) \oplus (n_{pi1} \oplus \beta_{pi}) \\
&<= ID_{pi} \oplus PW_{pi} \\
&PW_{pi} = ID_{pi} \oplus <
\end{aligned} \tag{10}
$$

여기서 *ID*는 생성과정에서 블록체인에 기록되기 때문에 블록에 참가할 수 있는 모두가 알 수 있다. 이 *ID*들을 하나하나 대입해 *PW*를 찾아낼 수 있다.

4-2 No Perfect Forward Secrecy

Perfect Forward Secrecy가 충족된다는 것은 프로토콜 상의 중요한 마스터키 중 하나가 노출되어도 이전 세션키를 알아낼 수 없다는 것이다. 그러나 본 스킴 에서는 변하지 않는 long-term 키 중 하나인 R값이 노출되었다고 가정했을 때 Perfect Forward Secrecy를 충족하지 못한다. 따라서 공격자가 미래에 long-term 키 중 하나인 R을 알아냈다고 가정 하면 해당 사용자와 서버 사이에서 사용된 모든 세션키를 계산할 수 있다. 그림3 은 공격자의 과거 세션키 계산 과정을 보여준다.

1. Adversary got DID_{pi} , D1, D2, D4, D5 in previous public channel
2. Adversary knew the long-term key R_{pi}
3. Adversary got ID_{pi} , SID_j from BlockChain

$\Rightarrow (n_{pi2} \parallel t_{pi-j}) = D1 \oplus ID_{pi}$
$\Rightarrow (n_j \parallel DID_{pi}^{new}) = (ID_{;pi} \parallel n_{pi2}) \oplus D4$
$\Rightarrow SK = h(ID_{pi} \parallel SID_j \parallel n_{pi2} \parallel n_j \parallel R_{pi})$

그림 3. No Perfect Forward Secrecy in Shamshad's Scheme
Fig. 3. No Perfect Forward Secrecy in Shamshad's Scheme

먼저 공격자는 사용자와 서버간의 이전 통신에서 *DID, D_1, D_2, D_4, D_5*를 가져올 수 있다. 그리고 공격자에게 long-term키 R이 노출되었다고 가정한다. 또한 공격자는 블록체인에서 사용자의 *ID, SID_j* 값을 가져올 수 있다. 그리고 공격자는 $(n_{pi2}||t_{p-j}) = D_1 \oplus ID$ 를 통해 n_{pi2} 와 t_{p-j}를 계산할 수 있고 다음으로 $(n_j \parallel DID_{newpi}) = (ID_{pi} \parallel n_{pi2}) \oplus D_4$를 통해 nj 와 새로운 *DID*를 알아낸다. 앞에서 알아낸 n_{pi2} 와 n_j를 사용하여 $SK = h(ID_{pi} \parallel SID_j \parallel n_{pi2} \parallel n_j \parallel R)$를 통해 세션키를 계산할 수 있다. long-term key R은 등록과정에서 PUF 함수를 통해 생성되어 서버의 DB에 저장되고 이후 바뀌지 않으므로 변하지 않는 키 이기 때문에 R을 알고 있다고 가정하면 이전의 세션키를 계산할 수 있다. 따라서 Perfect Forward Secrecy를 만족하지 못한다.

4-3 Insider Attack –Password Exposure

내부자 공격으로 인해 사용자의 패스워드(*PW*)가 노출될 수 있다. 일반적으로 패스워드는 서버에 저장될 때에도 패스워드 원본 그대로 가 아닌 해쉬값 으로 저장되기 때문에 서버의 관리자도 각 유저의 패스워드 원본을 알아낼 수 없다. 또한 Shamshad의 스킴 에서는 사용자의 *PW*가 단독으로 서버에게 전달되지도 않는다. *PW* 대신 *MPW*를 전달하는데 이를 통해 *PW*가 노출된다. 그림4는 공격자의 PW 계산과정을 보여준다.

1. Adversary get MPW_{pi} , $(n_{pi1} \oplus \beta_{pi})$ from Registration Phase

2. $MPW = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi}$
$\Rightarrow PW_{pi} = MPW \oplus n_{pi1} \oplus \beta_{pi}$

그림 4. Insider Attack in Shamshad's Scheme
Fig. 4. Insider Attack –1 in Shamshad's Scheme

먼저 내부자인 공격자는 등록과정에서 사용자에게서 전달받은 *MPW*와 $n_{pi1} \oplus B_{pi}$를 알 수 있다. 그리고 $MPW_{pi} = PW \oplus n_{pi1} \oplus B_{pi}$ 이므로 $MPW \oplus n_{pi1} \oplus B_{pi} = PW$를 통해 *PW*를 계산할 수 있다. 이로써 등록과정에서 받은 정보만 가지고 내부자가 해당유저의 P*W*를 알아낼 수 있다.

4-4 Insider Attack - User Impersonation

내부자가 유저를 가장하여 로그인에 성공할 수 있다. 스킴이 안전하다는 것을 증명하기 위해서는 공격자가 가장 중요한 키 하나를 제외한 모든 정보를 다 알아도 인증에 성공할 수 없어야 한다. 즉 어떤 환경에서도 '변하지 않는 키만 잘 숨겨놓으면 안전해야 한다. 그러나 본 프로토콜에서는 내부자가 사용자를 가장하여 인증에 성공 하고, 세션키를 계산할 수 있다. 그림 5에서는 내부자 공격에 의한 사용자 인증 및 세션키 계산 과정을 보여주고 있다.

1. Insider Adversary Knows t_{pi-j} , Id_{pi}, V_{pi-j} , BN_{pi} , DID_{pi}
2. And choose random Number n_{pi2}
3. Insider can get R_{pi} from Database

$\Rightarrow D1 = (n_{pi2} \parallel t_{pi-j}) \oplus ID_{pi}$
$\Rightarrow D2 = h(ID_{pi} \parallel n_{pi2} \parallel V_{pi-j} \parallel BN_{pi} \parallel t_{pi-j})$

4. U (Adversary) → {DID_{pi}, D1, D2} → S

5. U (Adversary) ← {D4, D5, C_{pi}} ← S

5. Then Adversary calculates $(n_j \parallel DID_{pi}^{new}) = (ID_{pi} \parallel n_{pi2}) \oplus D4$
6. So Adversary can calculate $SK = h(ID_{pi} \parallel SID_j \parallel n_{pi2} \parallel n_j \parallel R_{pi})$

그림 5. Insider Attack in Shamshad's Scheme
Fig. 5. Insider Attack – 2 in Shamshad's Scheme

먼저 내부자인 공격자는 이전 등록과정을 통해 t_{pi-j}, ID_{pi}, V_{pi-j}, B_{Npi}, DID_{pi}를 알 수 있다. 그리고 로그인 요청에 필요한 D_1은 $D_1 = (n_{pi2} \parallel t_{pi-j}) \oplus (ID_{pi})$ 이고 n_{pi2}는 세션마다 랜덤으로 생성되는 값이므로 랜덤한 숫자를 임으로 넣으면 된다. 그리고 $D_2 = h(ID_{pi} \parallel n_{pi2} \parallel V_{pi-j} \parallel B_{Npi} \parallel t_{pi-j})$를 계산하고 로그인 요청 {*DID, D_1, D_2*}를 보내고 응답 {D_4, D_5, C_{pi}}를 수신한다. 다음으로 $(n_j \parallel DID_{newpi}) = (ID_{pi} \parallel n_{pi2}) \oplus D_4$를 통해 n_j를 알아내고 $SK = h(ID_{pi} \parallel SID_j \parallel n_{pi2} \parallel n_j \parallel R_{pi})$ 를 통해 세션키를 계산할 수 있다. 결과적으로 공격자가 정상적인 사용자를 가장하여 로그인에 성공할 수 있다.

본 스킴 에서는 D_1과 D_2의 계산을 위해 유효한 패스워드와 B_{pi}(생체정보로 생성됨)가 필요하기 때문에 유저 사칭 공격에 안전하다고 하나 이처럼 내부자가 유저사칭 공격을 하는 경우 내부자의 정보만으로 로그인 정보 *DID*, D_1, D_2를 생성해 내고 세션키를 계산할 수 있다.

4-5 Login Interruption

공격자에 의해 정상적인 사용자의 로그인이 제한될 수 있다. 본 스킴의 로그인과정에서는 유저가 올바른 아이디, 비밀번호, 생체정보를 입력하면 등록과정에서 유저의 모바일 기기에 저장된 정보를 이용해 *DID*, D_1, D_2를 생성해 서버로 전송하게 된다. 이후 서버에서는 이 *DID*를 이용해 해당하는 유저의 정보를 데이터베이스에서 가져와서 D_1, D_2를통해 유저가 정상적인 사용자인지 확인하게 된다. 만약 정상적인 유저로 확인이 되면 새로운 *DID*를 생성하여 서버의 데이터베이스에서 업데이트하고, 세션 키를 생성하고, D_4, D_5를 유저에서 보내 유저의 기기에서도 새로운 *DID*로 업데이트 하게 되어 다음 로그인부터는 새로운 *DID*를 사용하게 되는 것 이다. 그러나 이후 따로 유저가 올바른 *SK*, *DID*등을 도출했는지 따로 검증하는 과정이 없기 때문에 로그인 방해 공격이 가능하다. 그림6은 공격자에 의한 로그인제한의 과정을 보여준다.

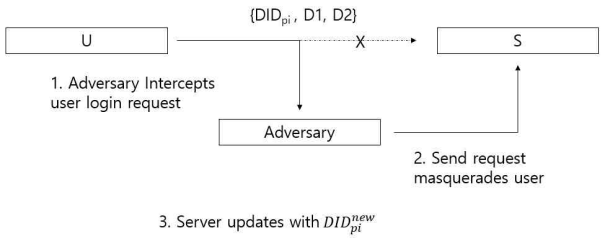

그림 6. Shamshad 등 스킴의 Login Interruption 분석
Fig. 6. Login Interruption in Shamshad's Scheme

먼저 스푸핑 공격 등을 통해 패킷을 가로챈 공격자가 유저를 가장하여 *DID*, D_1, D_2를 서버로 보낸다. 데이터 자체는 유효하기 때문에 서버에서는 새로운 *DID*를 생성하고, D_4, D_5를 보내주게 된다. 그 후 이 값을 유저에게 재전송 하지 않고 그대로 종료해버리면 로그인을 시도한 유저는 *DID*가 바뀐 것을 모르기 때문에 다음로그인부터는 본인의 *DID*를 전송해도 서버에서 해당하는 유저의 정보를 찾지 못하기 때문에 인증이 되지 않을 것이다. 그러므로 공격자에 의해 정상적인 사용자의 로그인이 제한될 수 있다.

4-6 Bit Mismatch

ShamShad의 스킴 에서는 XOR 연산이 많이 사용되는데 XOR 연산은 비트수가 일치해야 한다. 그러나 ShamShad 의 스킴 에서는 XOR 연산시 비트수가 안맞을 수밖에 없는 경우가 있다.

$$(n_{pi1} \| pri_{pi} \| t_{pi-j} \| BN_{pi}) = (ID_{pi} \| PW_{pi}) \oplus \alpha \oplus \beta_{pi}$$
$$MPW_{pi} = PW_{pi} \oplus n_{pi1} \oplus \beta_{pi} \quad (11)$$

여기서 공통적으로 β_{pi}가 사용되는데 아래수식에서 보면 β_{pi}의 비트수와 n_{pi}의 비트수와 PW_{pi}의 비트수는 같다는 것을 알 수 있다. 그러나 위의 수식에서 보면 $(ID_{pi} \| PW_{pi})$ 의 비트수와 β_{pi}의 비트수가 같다고 한다. $(ID_{pi} \| PW_{pi})$ 는 ID_{pi}와 PW_{pi} 비트를 이어 붙인 값인데 PW가 β_{pi}와 같다면 $(ID_{pi} \| PW_{pi})$와 β_{pi}가 같을 수가 없다. 그러므로 ShamShad의 스킴에는 비트수가 불일치해 XOR연산에 문제가 있을 수 있다.

Ⅴ. 결 론

Shamshad 등은 블록체인을 이용한 의료정보 저장 및 공유 스킴을 제안 하였고 본 논문은 Shamshad 등의 스킴의 취약점을 분석하여 오프라인 패스워드 추측공격, 내부자공격, 공격자에 의한 사용자 로그인 방해, 동작과정 비트수 불일치 의 취약점이 있는 것을 발견하였다.

참고문헌

[1] Mukherjee N, Neogy S, Chattopadhyay S. Big data in ehealthcare: challenges and perspectives. Chapman and Hall/CRC; 2019.

[2] Shen J, Shen J, Chen X, Huang X, Susilo W. An efficient public auditing protocol with novel dynamic structure for cloud data. IEEE Trans Inf Forensics Secur 2017; 12(10):2402–15.

[3] Shen J, Liu D, Bhuiyan MZA, Shen J, Sun X, Castiglione A. Secure verifiable database supporting efficient dynamic operations in cloud computing. IEEE Trans Emerg Top Comput 2017;8(2):280–90.

[4] Nakamoto S., Bitcoin A.. A peer-to-peer electronic cash system. Bitcoin–URL: htt ps://bitcoin.org/bitcoin.pdf2008.

[5] Swan M. Blockchain: blueprint for a new economy. "O'Reilly Media, Inc."; 2015.

[6] Shamshad S, Mahmood K, Kumari S, Chen CM et al (2020) A secure blockchain-based e-health records storage and sharing scheme. J Inform Secur Appl 55:102590

[7] Wazid M, Das AK, Kumar N, Vasilakos AV. Design of secure key management and user authentication scheme for fog computing services. Future Gener Comput Syst 2019;91:475–92.

[8] Wazid M, Das AK, Kumar N, Conti M, Vasilakos AV. A novel authentication and key agreement scheme for implantable medical devices deployment. IEEE J Biomed Health Inform 2017;22(4):1299–309.

[9] Zhou J, Cao Z, Dong X, Lin X, Vasilakos AV. Securing m-healthcare social networks: challenges, countermeasures and future directions. IEEE Wirel Commun 2013;20 (4):12–21.
[10] Porambage P, Ylianttila M, Schmitt C, Kumar P, Gurtov A, Vasilakos AV. The quest for privacy in the internet of things. IEEE Cloud Comput 2016;3(2):36–45.
[11] Wazid M, Das AK, Bhat V, Vasilakos AV. LAM-CIoT: lightweight authentication mechanism in cloud-based IoT environment. J Netw Comput Appl 2020;150: 102496.
[12] Zhou J, Cao Z, Dong X, Xiong N, Vasilakos AV. 4S: a secure and privacy-preserving key management scheme for cloud-assisted wireless body area network in m-healthcare social networks. Inf Sci 2015;314:255–76.
[13] Yan Z, Li X, Wang M, Vasilakos AV. Flexible data access control based on trust and reputation in cloud computing. IEEE Trans Cloud Comput 2015;5(3):485–98.
[14] Yang Y, Zheng X, Guo W, Liu X, Chang V. Privacy-preserving smart IoT-based healthcare big data storage and self-adaptive access control system. Inf Sci 2019; 479:567–92.
[15] Radha N, Karthikeyan S. A study on biometric template security. ICTACT J Soft Comput 2010;1(1):37–41.
[16] Aman MN, Chua KC, Sikdar B. Mutual authentication in IoT systems using physical unclonable functions. IEEE Internet Things J 2017;4(5):1327–40.
[17] Chen Y, Ding S, Xu Z, Zheng H, Yang S. Blockchain-based medical records secure storage and medical service framework. J Med Syst 2019;43(1):5.
[18] Zhang A, Lin X. Towards secure and privacy-preserving data sharing in e-health systems via consortium blockchain. J Med Syst 2018;42(8):140.

김종건(Jong-Geon Kim)

2016년 3월～현 재 : 인제대학교 컴퓨터공학부 학사과정

※관심분야 : 정보보호, 취약점분석

최윤성(Youn-Sung Choi)

2021년 3월~현 재 : 인제대학교 AI 융합대학 (산업보안전공) 조교수
2016년 3월~2020년 2월 : 호원대학교 사이버보안학과 조교수
2015년 8월 : 성균관대학교 전자전기 컴퓨터공학부 (공학박사)
2007년 8월 : 성균관대학교 전자전기 컴퓨터공학부 (공학석사)
2006년 2월 : 성균관대 정보통신공학부 (공학학사)

※관심분야 : 정보보호, 디지털포렌식, 산업보안