Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.6, Dec. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.6.999

영지식 증명을 활용한 복원 기능을 가진 검증 가능한 클라우드 기반의 개인 건강기록*

김 훈 기,[1†] 김 종 현,[1] 이 동 훈[2‡]
[1,2]고려대학교 정보보호대학원 (대학원생, 교수)

Verifiable Could-Based Personal Health Record with Recovery Functionality Using Zero-Knowledge Proof*

Hunki Kim,[1†] Jonghyun Kim,[1] Dong Hoon Lee[2‡]
[1,2]Graduate School of Information Security, Korea University
(Graduate student, Professor)

요 약

최근 개인 건강기록의 활용이 증가함에 따라, 개인 건강기록의 개인정보를 보호하는 암호 프로토콜에 대한 연구가 활발하게 이루어지고 있다. 현재 일반적으로 개인 건강기록은 암호화되어 클라우드에 외부 위탁되어 관리되고 있다. 하지만 이 방법은 개인 건강기록의 무결성을 검증하는데 제한적이며, 사용시 필수적으로 복호화가 필요하기 때문에 데이터 가용성이 떨어지는 문제점이 있다. 본 논문에서는 이 문제를 해결하기 위해 Redactable 서명기법과 영지식 증명을 사용하여 검증 가능한 클라우드 기반의 개인 건강기록 관리기법을 제안한다. 검증 가능한 클라우드 기반의 개인 건강기록 관리기법은 Redactable 서명기법을 사용함으로써 민감한 정보를 삭제하여 사생활(privacy)을 보존하면서 원본문서의 무결성 검증이 가능하며, 영지식 증명을 사용함으로써 원본문서의 삭제된 부분 외에는 삭제 및 수정되지 않았음을 검증 할 수 있다. 또한 Redact Recovery Authority를 통해 필요한 경우에만 삭제된 부분을 복원할 수 있도록 설계함으로써 기존의 관리기법보다 데이터의 가용성이 증가하도록 설계하였다. 그리고 제안한 기법을 활용한 검증 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안하고, 제안한 기법을 구현함으로써 효율성을 분석하였다.

ABSTRACT

As the utilize of personal health records increases in recent years, research on cryptographic protocol for protecting personal information of personal health records has been actively conducted. Currently, personal health records are commonly encrypted and outsourced to the cloud. However, this method is limited in verifying the integrity of personal health records, and there is a problem with poor data availability because it is essential to use it in decryption. To solve this problem, this paper proposes a verifiable cloud-based personal health record management scheme using Redactable signature scheme and zero-knowledge proof. Verifiable cloud-based personal health record management scheme can be used to verify the integrity of the original document while preserving privacy by deleting sensitive information by using Redactable signature scheme, and to verify that the redacted document has not been deleted or modified except for the deleted part of the original

Received(11. 20. 2020), Modified(12. 03. 2020), Accepted(12. 03. 2020)
* 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No. 2016-6-00599, 함수서명 설계기법 및 응용기술 연구).
† 주저자, hunki3882@korea.ac.kr
‡ 교신저자, donghlee@korea.ac.kr(Corresponding author)

document by using the zero-knowledge proof. In addition, it is designed to increase the availability of data than the existing management schemes by designing to recover deleted parts only when necessary through the Redact Recovery Authority. And we propose a verifiable cloud-based personal health record management model using the proposed scheme, and analysed its efficiency by implementing the proposed scheme.

Keywords: Verifiable Computation, zk-SNARK, Redactable Signature, Cloud-based Personal Health Record

I. 서 론

최근 의료기록은 기술의 발전으로 전자화 되어 관리가 되고 있다. 전자화된 의료기록을 발급기관에서 관리하고 있는 모델을 전자 건강기록이라고 하며 전자 건강기록의 확장된 개념으로 소유자 개인이 관리하고 있는 모델을 개인 건강기록이라고 한다[1]. 전자 건강기록은 환자의 접근통제가 이루어지지 않아 환자 동의 없이 사용될 수 있으며, 의료기관 간 전환이 쉽지 않다는 단점을 갖고 있다. 반면 개인 건강기록은 소유자가 중심이 되어 건강기록을 관리하기 때문에 전자 건강기록의 단점을 해결하였으며 보편적인 접근성으로 관심을 받고 있다.

개인 건강기록은 다양한 기관과 기기에 흩어져 있는 개인 건강정보를 통합하여 쉽게 확인하고 상호 공유하는 초기 모델에서 발전하여 자가 건강관리, 의료기관, 보험사와 연계한 질병예방 및 사후관리 등 유용한 서비스를 제공하는 모델로 발전하고 있다. 소유자는 개인 건강기록을 효율적으로 관리하기 위해 클라우드에 외부 위탁(outsourcing)하며 실제로 'Google Health'와 'Microsoft HealthVault'에 도입외어 사용되고 있다[2]. 개인 건강기록을 활용하기 위해서는 정보를 적극적으로 개방 및 공유해야 하지만 개인 건강기록이 갖고 있는 민감한 정보에 대한 사생활(privacy) 침해 문제가 뒤따르고 있어 개인정보를 지키기 위한 암호 기술을 필요로 한다.

영지식 증명(zero-knowledge proof)[3]은 어떠한 명제(statement)가 참임을 증명하고자 할 때 명제가 참이라는 사실 외에는 어떠한 정보도 노출하지 않는 암호학적 증명기법이다. 그 중 zk-SNARK(zero-knowledge Succinct Non-interactive ARgument of Knowledge)[4]는 기존의 영지식 증명을 간결하고 비상호적인 환경에서 적용한 기법이다. Redactable 서명[4,5]은 서명된 메시지의 일부를 삭제하더라도 서명자의 서명검증이 유효한 기법이다. 따라서 민감한 정보를 포함한 서명된 문서에 대해서 Redactable 서명기법으로 민감한 정보를 삭제하고 삭제한 부분에 대한 영지식 증명을 생성하면 문서 사용자는 영지식 증명 검증과정을 통해 삭제된 부분 외에는 변경되지 않았음을 확인하고 문서의 삭제되지 않은 부분을 사용할 수 있다[7].

본 논문에서는 Redactable 서명으로 삭제된 부분을 공개하지 않고 영지식 증명으로 검증할 수 있는 Chabanee등의 아이디어[7]에 기반을 두어 삭제된 문서를 복원할 수 있는 기능을 추가한 기법을 제안한다. 또한 이 기법을 활용하여 민감한 정보를 보호하며 검증이 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안한다.

1.1 기여도

본 논문에서는 zk-SNARK[4]와 Redactable 서명[4,5]으로 복원 기능을 가진 검증 가능한 삭제된 문서기법을 제안하고 이를 활용한 검증 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안한다.

기존의 연구[7]에 기반 하여 삭제된 문서를 복원하는 기능을 추가한 기법을 제안한다. 기존 기법에서는 데이터 사용자가 문서의 원본을 필요로 하지만 원본문서 소유자에게 접근이 불가능할 경우 원본문서를 사용할 수 없게 된다. 이를 해결하기 위해 소유자는 삭제된 문서와 원본문서의 암호문으로 증명을 생성하고 사용자가 원본문서가 필요할 경우 신뢰기관(trusted-authority)인 Redact Recovery Authority(RRA)를 통해 암호문을 복호화 함으로써 원본 문서를 복원할 수 있도록 설계하였다.

또한 제안하는 기법을 활용하여 검증 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안한다. 개인 건강기록의 소유자는 발급 기관으로부터 서명된 개인 건강기록의 일부를 삭제하도록 허용되어 민감한 정보를 삭제하고 삭제한 부분과 개인 건강기록의 암호문으로 zk-SNARK 증명을 생성하여 특정 경우에 RRA를 통해 복원될 수 있음을 동의하며 클라우드에 외부 위탁한다. 클라우드로부터 민감한 정보가 삭제된 개인 건강기록을 사용하는 데이터 소비자는 업로드한 소유자와 발급기관이 오프라인(off-line)이기

때문에 개인 건강기록의 원본이 필요할 경우 RRA를 통해 복원 받아 원본을 사용하게 된다.

마지막으로 제안하는 기법을 구현한 후 증명키, 검증키의 크기와 증명 생성(prove) 및 증명 검증(verify)에 소요되는 시간을 측정하여 본 논문에서 제안하는 기법의 효율성을 실험적으로 확인하였다.

1.2 관련 연구

개인 건강기록의 개인정보를 보호하기 위한 암호 기술의 다양한 연구가 이루어지고 있다. 개인 건강기록은 일반적으로 클라우드 플랫폼에 위탁되어 소유자가 데이터를 관리하는 부담을 덜어주며 의료 데이터의 가용성을 향상시킨다. 하지만 개인 건강기록을 외부에 위탁하는 것은 허가받지 않은 당사자에게 건강 정보를 유출할 위험이 높기 때문에 상당한 개인정보 보호 침해 우려를 야기한다.

이를 해결하기 위해 2013년 Wang 등은 Bethencourt 등이 제안한 속성기반 암호[8]를 사용한 클라우드 기반의 개인 건강기록 모델[9]를 설계하였다. 2014년에는 Gondkar등이 클라우드 기반의 개인 건강기록을 관리하기 위한 속성기반 암호[10]을 제안하였으며 2017년 Rao등과 2018년 Deng등은 속성기반 서명암호(signcryption)를 사용한 기법[11,12]를 제안하였다.

이 기법들은 모두 암호화 기법을 사용하여 클라우드 기반의 개인 건강 기록의 개인정보를 보호한다. 하지만 Wang 등이 제안한 기법[9]와 Gondkar 등이 제안한 기법[10]은 클라우드로부터 사용되는 개인 건강기록은 건강기록 발급기관으로부터의 무결성을 검증 할 수 없다. 또한 Rao등과 Deng등의 기법[11,12]는 서명암호를 사용했기 때문에 무결성을 검증할 수 있지만 클라우드로부터 개인 건강 기록을 사용하기 위해서는 복호화 된 원본을 확인하거나 확인하지 못하도록 제한되어 설계되었기 때문에 데이터의 활용이 제한된다. 클라우드 기반의 개인 건강기록 기법에 대한 비교는 다음 Table 1.과 같다.

Table 1. Cloud-based PHR schemes

schemes	privacy	integrity	selective data access
[9,10]	O	X	X
[11,12]	O	O	X
ours	O	O	O

따라서 본 논문에서는 이를 해결하기 위해 Redactable 서명기법을 사용함으로써 소유자가 문서의 일부를 삭제하도록 허용하여 필요한 정보만 공유할 수 있어 개인 건강기록의 안전하고 효율적인 관리가 가능하도록 하였으며, 데이터 소비자는 원본문서의 서명과 영지식 증명을 통해 그들이 사용하는 데이터가 서명된 문서에 기초하였으며 삭제된 부분 외에는 변경되지 않았음을 검증할 수 있으며, 원본문서가 필요할 경우 RRA를 통해 복원하여 사용함으로써 의료 데이터의 활용도를 높일 수 있도록 제안하였다. 또한 zk-SNARK를 사용하였기 때문에 증명의 크기가 간결하며 클라우드에서 수행되는 검증시간이 빠르게 설계되었다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 본 논문에서 제안한 기법의 배경지식에 대해 설명하고, Ⅲ장에서는 Verifiable Document Redacting 기법, Ⅳ장에서는 제안 기법 정의와 안전성 정의에 대해 설명하고, Ⅴ장에서는 제안한 기법을 활용한 검증가능한 클라우드 기반의 개인 건강기록 모델을 제안하고, Ⅵ장에서는 제안 기법의 구현결과를 분석한다. 마지막으로 Ⅶ장에서는 결론을 맺는다.

Ⅱ. 배경지식

2.1 Verifiable Computation

클라우드 컴퓨팅 기술이 발전하면서, 상대적으로 연산능력이 부족한 사용자가 클라우드를 통해 연산 작업을 위탁하여 효율적으로 연산하는 하는 기법들이 제안되었다. 그중 Verifiable Computation[13]은 증명자가 연산 결과와 증명(proof)을 만들면 검증자가 증명을 통해 연산하는 시간보다 빠르게 연산이 잘되었는지 검증하는 기법이다.

2.1.1 알고리즘

Verifiable Computation 기법은 $Setup$, $Prove$, $Verify$ 세 가지 알고리즘으로 구성된다. $y=f(x)$ 연산을 증명하고자 할 때, f는 유한 체 F상에서 산술회로를 λ는 보안 상수를 나타낸다.

- $Setup(1^{\lambda},f)\rightarrow(EK_f,VK_f)$: $Setup$알고리즘은 보안 상수 λ와 함수 f를 입력받아 증명 π를 만

드는데 사용되는 증명키 EK_f, 연산검증에 사용되는 증명 검증키 VK_f를 출력한다. EK_f와 VK_f는 공개키로 사용되며 함수 f에 의존하여 생성 후 재사용 될 수 있다.

- $Prove(EK_f,x)\rightarrow(y,\pi)$: $Prove$알고리즘은 증명키 EK_f와 함수 f의 입력값 x를 입력받아 연산 결과 y와 증명 π를 출력한다.
- $Verify(VK_f,x,y,\pi)\rightarrow(0,1)$: $Verify$알고리즘은 증명 검증키 VK_f, 입력값 x, 연산 결과 y와 증명 π를 입력받아 증명 π를 통해 연산이 잘 되어있는지 검증 후 검증 결과가 참일 경우 1, 그렇지 않으면 0을 출력한다.

2.2 영지식 증명(zero-knowledge proof)

영지식 증명[3]이란 증명자가 검증자에게 어떠한 명제(statemt)가 참이라는 것을 증명하고자 할 때 명제의 참/거짓 이외의 숨기고자 하는 정보(witness)를 공개하지 않고 증명하는 증명방법이다.

2.2.1 알고리즘

영지식 증명의 알고리즘은 $Setup$, $Prove$, $Verify$ 세 가지 알고리즘으로 구성된다. 증명자가 $f(w,x)=y$ 연산을 증명하고자 할 때, x는 증명하고자 하는 명제를 w는 숨기고자 하는 정보를 나타낸다.

- $Setup(1^\lambda,f)\rightarrow(EK_f,VK_f)$: $Setup$알고리즘은 보안 상수 λ와 함수 f를 입력받아 증명 π를 만드는데 사용되는 증명키 EK_f, 연산검증에 사용되는 증명 검증키 VK_f를 출력한다. EK_f와 VK_f는 공개키로 사용된다.
- $Prove(EK_f,x,w)\rightarrow(y,\pi)$: $Prove$알고리즘은 증명키 EK_f, 명제 x와 정보 w를 입력받아 연산 결과 y와 증명 π를 출력한다.
- $Verify(VK_f,x,y,\pi)\rightarrow(0,1)$: $Verify$알고리즘은 증명 검증키 VK_f, 명제 x, 연산 결과 y와 증명 π를 입력받아 증명 π를 통해 연산이 잘 되어 있는지 검증 후 검증 결과가 참일 경우 1, 그렇지 않으면 0을 출력한다.

2.2.2 특성

영지식 증명 알고리즘이 만족하는 특성은 다음과 같다.

- 완전성(completeness) : 명제가 참이면, 정직한 증명자는 검증을 통과한다.
- 건전성(soundness) : 명제가 거짓이면, 어떠한 부정직한 증명자라도 검증을 통과할 수 없다.
- 영지식성(zero-knowledge) : 명제가 참이면, 참이라는 사실 외의 정보는 노출되지 않는다.

2.3 zk-SNARK

zk-SNARK(zero-knowledge Succinct Non-interactive ARgument of Knowledge)[4]는 영지식 증명의 하나로써 기존의 영지식 증명을 좀 더 간결(succinct)하고 비상호적인 환경(non-Interactive)에서 적용한 기법이며 증명의 크기가 작고 검증에 소요되는 시간이 짧다는 장점이 있다.

- Succinct : 증명 π의 크기가 간결하며 검증이 쉽다.
- Non-Interactive : 영지식 증명이 이루어 질 때 증명자와 검증자가 상호 작용을 하지 않고 증명이 이루어진다.
- Argument of Knowledge : zk-SNARK는 계산적으로 확실한 것으로 간주되어 증명자의 컴퓨팅 능력이 제한적이라고 가정했을 때, 부정직한 증명자가 시스템을 속일 수 있는 확률이 매우 낮다.

2.3.1 알고리즘

zk-SNARK의 알고리즘은 $Setup$, $Prove$, $Verify$, $SmProve$ 네 가지 알고리즘으로 구성된다. 이때, R은 명제 x와 정보 w의 관계(relation)를 σ는 공용변수(common reference string, crs)를 τ는 공용변수생성 과정에서 이용된 트랩도어(trapdoor)를 나타낸다.

- $Setup(R)\rightarrow(\sigma,\tau)$: $Setup$알고리즘은 관계 R을 입력받아 공용변수 σ를 출력한다.

- $Prove(R,\sigma,x,w)\rightarrow(\pi)$: $Prove$알고리즘은 관계 R, R에 대한 공용변수 σ, 명제 x, 정보 w를 입력받아 증명 π를 출력한다.
- $Verify(\sigma,x,\pi)\rightarrow(0,1)$: $Verify$알고리즘은 공용변수 σ, 명제 x, 증명 π를 입력으로 받아 증명의 검증 결과가 참일 경우 1, 그렇지 않으면 0을 출력한다.
- $SmProve(\sigma,\tau,x)\rightarrow\pi$: $SmProve$시뮬레이션은 공용변수 σ, 트랩도어 τ, 명제 x를 입력받아 증명 π를 출력한다.

2.3.2 특성

zk-SNARK 알고리즘이 만족하는 특성은 다음과 같다.

- 완전한 완전성(perfect completeness) : 올바른 명제 x가 주어졌을 때, 정보 w를 알고 있는 증명자는 $Verify$알고리즘을 항상 통과할 수 있음을 의미하며 다음의 확률을 만족한다.

$$\Pr\left[\begin{matrix}Setup(R)\rightarrow(\sigma,\tau);Prove(R,\sigma,x,w)\rightarrow(\pi):\\ Verify(R,\sigma,x,\pi)=1\end{matrix}\right]$$

- 계산적 지식 건전성(computational knowledge soundness) : 다항식 시간 익스트렉터(extractor) E가 존재할 때, 모든 다항식 시간 공격자 A가 $Verify$알고리즘을 통과할 경우 해당 명제 x가 참이고 증명자가 정보 w를 알고 있을 확률은 무시할(negligible)만하다. 이는 다음의 확률을 만족한다.

$$\Pr\left[\begin{matrix}Verify(R,\sigma,x,\pi)=1;(x,\pi)\notin R\\ A(R,\sigma,z)\rightarrow(x,\pi);E(R,\sigma,z)\rightarrow w\end{matrix}\right]\leq legl(\lambda)$$

- 완벽한 영지식성(perfect zero-knowledge) : 모든 다항식 시간 공격자 A는 $Prove$알고리즘을 통해 제대로 된 명제 x와 정보 w로 만들어진 증명 π와 $SmProve$시뮬레이션으로 정보 w를 없이 만들어진 증명 π를 구분할 수 없음을 의미하며 다음의 확률을 만족한다.

$$\Pr\left[\begin{matrix}Setup(R)\rightarrow(\sigma,\tau);Prove(R,\sigma,x,w)\rightarrow\pi:\\ A(R,\tau,\pi)=1\end{matrix}\right]=\Pr\left[\begin{matrix}Setup(R)\rightarrow(\sigma,\tau);SmProve(R,\tau,x)\rightarrow\pi:\\ A(R,\tau,\pi)=1\end{matrix}\right]$$

- 간결성(succinctness) : 증명의 크기는 $O_\lambda(1)$이며 검증에 소요되는 시간은 $O_\lambda(|x|)$이다.

2.4 Redactable 서명

Redactable 서명[4,5]는 서명자의 서명키로 생성된 메시지, 서명 쌍에 대하여 삭제자가 메시지의 일부를 삭제하더라도 서명키의 검증키로 삭제된 문서의 무결성을 확인할 수 있는 기법이다.

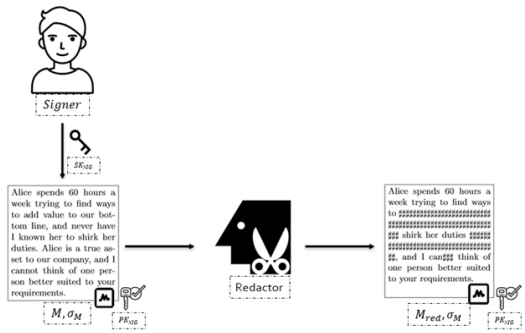

Fig. 1. Redacatble Signature

2.4.1 알고리즘

Redacatble 서명기법은 Gen, $Sign$, $Redact$, Ver 네 가지 알고리즘으로 구성된다. 이때, λ는 보안 상수를 $MOD\in\{1,2,\cdots,n\}$는 삭제할 블록의 색인번호(index number)를 $M_{red}=(m_1^{red},m_2^{red},\cdots,m_n^{red})$는 삭제된 메시지를 나타내며 $Sign$알고리즘의 서명기법은 포괄적(generic)으로 설계한다.

- $Gen(1^\lambda)\rightarrow(SK_{SIG},PK_{SIG})$: Gen알고리즘은 보안 상수 λ를 입력받아 서명생성에 사용되는 서명키 SK_{SIG}와 서명검증에 사용되는 검증키 PK_{SIG} 쌍을 출력한다.
- $Sign(M,SK_{SIG})\rightarrow\sigma_M$: $Sign$알고리즘은 n개의 블록으로 이루어진 메시지 $M=(m_1,m_2,\cdots,m_n)$과 서명키 SK_{SIG}를 입력 받아 서명 σ_M을 출력한다.
- $Redact(PK_{SIG},M,\sigma_M,MOD)\rightarrow(M_{red},\sigma_{M_{red}})$:

  $Redact$알고리즘은 메시지를 삭제하는 삭제자에 의해 이루어진다. 검증키 PK_{SIG}, 메시지 M, 서명 σ_M, 색인번호 MOD를 입력받아 MOD에 해당하는 메시지 블록을 삭제하여 삭제된 메시지

M_{red}와 삭제된 메시지의 서명 $\sigma_{M_{red}}$를 출력한다.

- $Verify(PK_{SIG}, M_{red}, \sigma_{M_{red}}) \rightarrow (0,1)$: Ver알고리즘은 검증키 PK_{SIG}, 삭제된 메시지 M_{red}, 서명 $\sigma_{M_{red}}$를 입력받아 검증키 PK_{SIG}로 삭제된 메시지의 서명을 검증 후 검증 결과가 참일 경우 1, 그렇지 않으면 0을 출력한다.

III. Verifiable Document Redacting

3.1 Verifiable Document Redacting 정의

2017년 Chabanee등은 zk-SNARK[4]를 기반으로 Redactable 서명[5,6] 기법을 통해 인증 받은 문서의 민감한 정보를 노출하지 않고도 검증할 수 있는 VDR(Verifiable Document Redacting)기법[7]을 제안하였다.

VDR기법은 세 개의 개체(Entity)에 의해 수행된다. 첫 번째 개체는 인증된 문서를 발행하는 DI(Document Issuer), 두 번째 개체는 인증된 문서의 민감한 부분을 삭제하는 Cl(Client), 세 번째 개체는 삭제된 문서를 검증하고 사용하는 DU(Document User)가 있다. DI가 Redactable 서명기법으로 서명한 문서를 Cl에게 주면 Cl은 Redactable 서명기법의 삭제자가 되어 문서의 민감한 정보를 삭제하고 공개하지 않기 위해 영지식 증명을 사용하여 삭제된 부분을 정보(witness)로 하여 증명(proof)을 만든다. 문서를 사용하는 DU는 서명을 통해 DI로부터 문서가 발행되었음을 검증하고 Cl가 생성한 증명을 통해 삭제된 부분을 공개하지 않고 삭제된 문서가 유효하다는 것을 검증할 수 있다. 기존의 Redactable 서명은 누구나 메시지에 대한 삭제를 할 수 있던 점과 다르게 이 기법에서는 삭제된 민감한 정보를 공개하지 않기 위해 삭제자는 Cl로 한정하여 설계되었다.

Chabanee등은 zk-SNARK를 사용했기 때문에 VDR기법으로 기존의 PhotoProof기법[14]의 증명 크기를 줄여 검증을 빠르게 할 수 있도록 최적화하여 구현하였다.

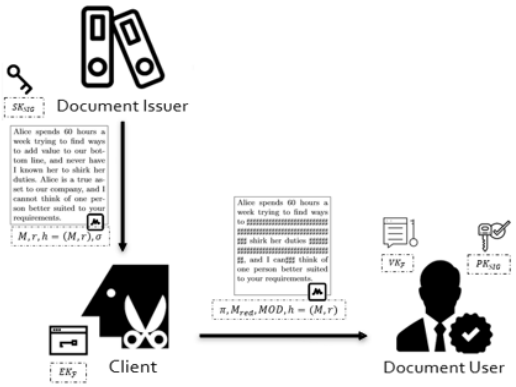

Fig. 2. Verifiable Document Redacting

3.1.1 알고리즘

VDR(Verifiable Document Redacting)기법은 Redactable 서명[5,6] 기법의 Gen, $Sign$, Ver알고리즘과 zk-SNARK[4] 기법의 $Setup$, $Prove$, $Verify$알고리즘을 사용하며 $KeyGen$, $Authent$, $Redact$, $DocVerify$ 네 개의 알고리즘으로 구성된다. 이때, H는 해쉬함수를 나타낸다.

- $KeyGen(1^{\lambda}, \mathbb{F}) \rightarrow (SK_{SIG}, PK_{SIG}, EK_{\mathbb{F}}, VK_{\mathbb{F}})$: $KeyGen$알고리즘은 보안 상수 λ와 유한 체 F_p 상의 산술 회로 $\mathbb{F}$를 입력받아 Gen알고리즘으로 생성된 서명키 SK_{SIG}, 검증키 PK_{SIG}와 $Setup$알고리즘으로 생성된 증명키 $EK_{\mathbb{F}}$, 증명 검증키 $VK_{\mathbb{F}}$를 출력하여 총 네 개의 키를 출력한다. PK_{SIG}, $VK_{\mathbb{F}}$는 공개키이며 SK_{SIG}는 DI의 개인키, $EK_{\mathbb{F}}$는 Cl의 개인키가 된다.
- $Authent(M, SK_{SIG}) \rightarrow (r, h, \sigma)$: $Authent$알고리즘은 원본 메시지 M과 서명키 SK_{SIG}를 입력받아 임의의 난수 r, 해쉬값 $h = H(M \| r)$와 h의 서명값 $\sigma = Sign(h, SK_{SIG})$를 출력한다.
- $Redact(M, h, r, \sigma, EK_{\mathbb{F}}) \rightarrow (M_{red}, MOD, h, \sigma, \pi)$: $Redact$알고리즘은 Cl에 의해 이루어지며 원본 메시지 M, 해쉬값 h, 난수 r, 서명값 σ, 증명키 $EK_{\mathbb{F}}$를 입력받는다. Cl은 삭제자로써 삭제할 메시지의 색인번호(index number) $i \in MOD$에 해당하는 메시지 블록을 $m_i = \#$로 삭제하고 M과 r을 정보로 h, M_{red}, MOD를 명제(statement)로 설정하여 $Prove$알고리즘으로 π를 생성한다.

$$\begin{cases} Prove(([M,r],h,M_{red},MOD),EK_{\mathbb{F}}) \rightarrow \pi \\ witness : M,r \\ statement : h,M_{red},MOD \end{cases}$$

따라서 삭제된 메시지 M_{red}, 색인번호 MOD, 해쉬값 h, 서명값 σ, $Prove$알고리즘으로 생성된 증명 π를 출력한다.

- $DocVerify(M_{red}, MOD, h, \sigma, \pi, VK_{\mathbb{F}}, PK_{SIG}) \rightarrow (0,1)$: $DocVerify$알고리즘은 DU에 의해 이루어지며 삭제된 메시지 M_{red}, 색인번호 MOD, 해쉬값 h, 서명값 σ, 증명 π, 증명 검증키 $VK_{\mathbb{F}}$, 서명 검증키 PK_{SIG}를 입력받아 Ver알고리즘으로 서명값 σ에 대한 검증이 통과할 경우 $Verify$알고리즘으로 증명 π에 대한 검증을 하여 검증 결과가 참일 경우 1, 그렇지 않으면 0을 출력한다.

IV. Verifiable Document Redacting with Recovery Functionality

본 장에서는 제안하는 기법의 알고리즘을 정의하며, 알고리즘이 만족하는 안전성을 정의한다.

4.1 Verifiable Document Redacting with Recovery Functionality 정의

Verifiable Document Redacting with Recovery Functionality(VDRRF)기법은 4개의 개체(entity)에 의해 수행된다. 첫 번째 개체는 인증된 문서를 발행하는 DI(Document Issuer), 두 번째 개체는 인증된 문서의 민감한 부분을 삭제하며 삭제된 메시지를 복원하기 위한 암호문을 생성하는 Cl(Client), 세 번째 개체는 삭제된 문서를 검증하고 사용하는 DU(Document User), 마지막으로 Cl이 생성한 암호문을 복호화 함으로써 원본문서를 복원하는 신뢰기관(trusted-authority) Redact Recovery Authority(RRA)가 있다. 이때, 신뢰기관 RRA는 공공기관이며 민감한 정보를 삭제한 Cl은 RRA의 공개키로 암호화하며 RRA가 원본문서를 특정 경우에 한에 복원하는 것에 정책적으로 동의함을 가정한다.

DI가 자신의 서명키로 서명하여 인증된 문서를 Cl에게 주면 Redactable 서명기법의 삭제자가 되어 문서의 민감한 정보를 삭제하고 삭제된 부분을 RRA의 공개키로 암호화하여 암호문을 만든다. 이때, 삭제된 부분을 공개하지 않기 위해 영지식 증명을 사용하여 삭제된 부분을 정보(witness)로 하여 증명(proof)을 만든다. 문서를 사용하는 DU는 서명을 통해 DI로부터 문서가 발행되었음을 검증하고 Cl가 생성한 증명을 통해 삭제된 부분을 공개하지 않고 삭제된 문서가 유효하며 암호문이 원본문서로부터 생성되었다는 것을 검증할 수 있다. 이때 증명은 zk-SNARK기법으로 만들어졌기 때문에 증명의 크기가 간결하여 검증이 빠르게 이루어진다. RRA는 DU의 요청이 Cl이 동의한 특정 경우에 해당될시 암호문을 자신의 개인키로 복호화 하여 삭제된 부분을 DU에게 제공한다.

문서를 사용하는 DU는 삭제된 부분외의 문서는 사용할 수 있지만 삭제된 부분을 사용하기 위해서는 DI 또는 Cl를 통해 원본문서를 받아야 한다. 하지만 DI 또는 Cl이 원본문서를 제공할 수 없는 오프라인(off-line)이거나 DU가 다수의 문서를 사용할 경우 원본문서를 사용하는데 있어 제약이 생기게 된다. 이를 해결하기 위해 이 기법에서는 RRA를 통한 복원 기능을 추가함으로써 데이터의 가용성 증가를 제공한다.

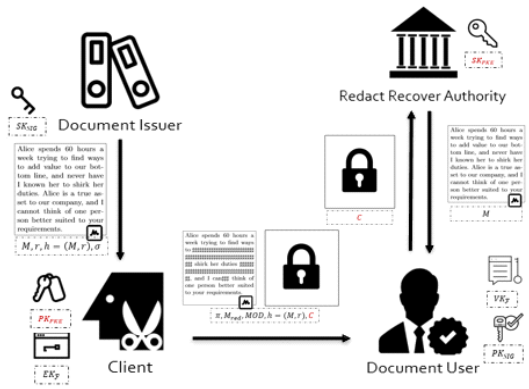

Fig. 3. Verifiable Document Redacting with Recovery Functionality

4.1.1 알고리즘

VDRRF기법은 Redactable 서명[5,6] 기법의 Gen, $Sign$, Ver알고리즘, zk-SNARK[4] 기법의 $Setup$, $Prove$, $Verify$알고리즘, Public Key Encryption(PKE)기법의 Key, Enc, Dec알고리즘을 사용하며 $KeyGen$, $Authent$, $Redact$, $DocVerify$, $Recovery$ 다섯 개의 알고리즘으로 구

성된다. 이때 *Redact*알고리즘에서 생성되는 메시지 M에 대한 암호문 C는 복원 기능을 사용하기 위해 RRA의 공개키로 생성되며 사용되는 공개키 암호시스템(PKE)은 포괄적(generic)으로 설계 된다. 자세한 알고리즘은 Fig. 2-2.와 같다.

- *KeyGen*알고리즘은 서명과 zk-SNARK, 암호화에 사용되는 키를 생성한다. *Gen*알고리즘은 DI에 의해 *Setup*알고리즘은 Cl 의해 *Key*알고리즘은 RRA에 의해 이루어진다. PK_{SIG}, $VK_{\mathbb{F}}$, PK_{PKE}는 공개키이며 SK_{SIG}는 DI의, $EK_{\mathbb{F}}$는 Cl의, SK_{PKE}는 RRA의 개인키가 된다.
- *Authent*알고리즘은 DI에 의해 이루어지며 원본 메시지 M과 임의의 난수 r의 해쉬값 $h=H(M\|r)$를 계산하여 h에 대한 서명 $\sigma=Sign(h,SK_{SIG})$를 만든다.
- *Redact*알고리즘은 Cl에 의해 이루어지며 서명값 σ가 *Ver*알고리즘으로 검증이 통과 됐을 경우 수행된다. 삭제자로써 원본 메시지 블록에서 삭제될 부분을 정해 색인번호(index number) $i\in MOD$를 정의하고 MOD에 해당하는 메시지 블록은 $m_i=\#$로 삭제한다. 삭제된 메시지를 복원하기 위해 RRA의 공개키로 삭제된 메시지 블록 m_i들의 집합 M_{mod}와 임의의 난수 s에 대한 암호문 C를 만든다. M,r,s를 숨기고자하는 정보로 h, M_{red}, MOD, C를 증명하고자 하는 명제(statement)로 하여 *Prove*알고리즘으로 증명 π를 생성한다. 따라서 M,r,s를 정보로 함으로써 증명 π에 대한 검증 시 원본 메시지의 삭제된 블록이 공개되지 않고 검증이 이루어지게 된다.
- *DocVerify*알고리즘은 DU에 의해 이루어지며 *Ver*알고리즘으로 서명값 σ에 대한 검증을 한 뒤 서명 검증이 통과할 경우 증명 π에 대한 검증이 이루어진다. DU는 증명 π를 통해 삭제된 메시지 블록을 확인하지 않고 h는 M과 r의 해쉬값이며 MOD외의 메시지블록은 수정되지 않았으며 C는 M_{mod}와 s로 생성된 암호문임을 검증할 수 있다. 두 검증이 모두 참일 경우 검증을 통과하며 그렇지 않을 경우 검증에 실패한다.
- *Recovery*알고리즘은 RRA에 의해 이루어지며 DU가 삭제된 메시지 M_{red}에 대한 복원을 요청할 경우 암호문 C를 개인키 SK_{PKE}로 복호화 하

$KeyGen(1^{\lambda},\mathbb{F})$:
$(SK_{SIG},PK_{SIG}) <- Gen(1^{\lambda})$
$(EK_{\mathbb{F}},VK_{\mathbb{F}}) <- Setup(1^{\lambda},\mathbb{F})$
$(PK_{PKE},SK_{PKE}) <- Key(1^{\lambda})$
return SK_{SIG}, PK_{SIG}, $EK_{\mathbb{F}}$, $VK_{\mathbb{F}}$, SK_{PKE}, PK_{PKE}

$Authent(M,SK_{SIG})$:
$r \xleftarrow{\$} \{0,1\}^{128}$
$h=H(M\|r)$
$\sigma <- Sign(h,SK_{SIG})$
return h, r, σ

$Redact(M,h,r,\sigma,EK_{\mathbb{F}})$:
$d \leftarrow Ver(h,\sigma,PK_{SIG})$
if $d=0$, the abort
Else :
define the set $MOD \in \{1,2,\cdots,n\}$
define M_{red}, $\forall i \in MOD$, $m_i^{red}=\#$,
$m_i^{red} \in M_{\mathrm{mod}}$
$s \xleftarrow{\$} \{0,1\}^{128}$
$C \leftarrow Enc(M_{\mathrm{mod}},s,SK_{PKE})$
$\pi \leftarrow Prove([M,r,s],h,M_{red},MOD,C),EK_{\mathbb{F}})$
witness : M,r,s
statement : h,M_{red},MOD,C
return M_{red}, MOD, h, σ, π, C

$DocVerify(M_{red},MOD,h,\sigma,\pi, VK_{\mathbb{F}},PK_{SIG},C)$:
$d \leftarrow Ver(h,\sigma,PK_{SIG})$
$d \leftarrow d \times Verify(\pi,(M_{red},h,MOD,C),VK_{\mathbb{F}})$
if $d \equiv 1$ return d
if $d \equiv 0$ return $\perp$

$Recovery(C,SK_{PKE})$:
$M_{\mathrm{mod}} \leftarrow DEC(C,SK_{PKE})$
return M_{mod}

Fig. 4. Specific of Verifiable Document Redacting with Recovery Functionality

여 삭제된 메시지 블록들의 집합 M_{mod}를 복원한다.

4.2 Verifiable Document Redacting with Recovery Functionality 안전성 정의

본 논문에서 제안하는 Verifiable Document Redacting with Recovery Functionality(VDRRF)기법의 안전성은 다음 2가지 안전성을 만족하며, Redactable 서명기법 [5,6]의 안정성을 적용한다.

1) Privacy : 삭제된 메시지, 증명, 암호문만 갖고 있는 확률적인 다항시간의 공격자는 삭제된 부분의 원본을 복원할 수 없다.
2) Unforgeability : 원본 메시지를 갖고 있지 않는 확률적인 다항시간의 공격자는 삭제된 문서와 증명을 만들 수 없다.

4.2.1 Privacy

- Privacy : 만약 모든 확률적인 다항시간의 공격자 A로부터 아래의 실험 Leak의 출력이 1일 확률이 무시할 만큼 $\frac{1}{2}$에 가깝다면 VDRRF기법 ($KeyGen$, $Authent$, $Redact$, $DocVerify$, $Recovery$)은 Privacy를 만족한다.

  실험 Leak :
  $\{0,1\}$중 임의의 값을 b라 하고, 색인번호 $i \in MOD$와 $\forall j \neq i$에 대해 $m_j^0 = m_j^1$이고 $m_i^0 \neq m_i^1$를 만족하는 (M^0, M^1, i)가 A에게 주어졌을 때, $Authent$알고리즘과 $Redact$알고리즘을 수행하는 오라클(oracle) $O^{Authent/Redact}$을 통해 M^b에 대한 $(M_{red}^b, h_b, C_b, \sigma_b, \pi_b)$를 생성한다. A는 $(PK_{SIG}, VK_{\mathbb{F}}, PK_{PKE}, M_{red}^b, h_b, C_b, \sigma_b, \pi_b)$를 통해 b^*를 정하고 만약 $b^* = b$일 경우 실험 Leak 는 1을 출력한다.

  따라서 공격자의 어드벤티지(advantage)를 $\mathrm{Adv}_{\mathrm{Leak}}^{A} = \left| \Pr[\mathrm{Leak} = 1] - \frac{1}{2} \right|$이라 정의하고, 만약 모든 확률적인 다항시간의 공격자로부터 $\mathrm{Adv}_{\mathrm{Leak}}^{A}$값이 무시할 만큼 작다면 VDRRF기법은 Privacy를 만족한다.

4.2.2 Unforgeability

- Unforgeability : 만약 모든 확률적인 다항시간의 공격자 A로부터 아래의 실험 Forge의 결과가 1일 확률이 무시할 만큼 작다면 VDRRF기법 ($KeyGen$, $Authent$, $Redact$, $DocVerify$, $Recovery$)은 Unforgeability를 만족한다.

  실험 Forge(λ) :
  $KeyGen(\lambda)$알고리즘으로 $PK_{SIG}, VK_{\mathbb{F}}, PK_{PKE}$ $SK_{SIG}, EK_{\mathbb{F}}, SK_{PKE}$를 생성하고, $Authent$알고리즘과 $Redact$알고리즘을 수행하는 오라클(oracle) $O^{Authent/Redact}$을 통해 $i = 1, \cdots, q$를 만족하는 M^i에 대해 $(M_{red}^i, MOD_i, \sigma_i, \pi_i)$를 생성하고 A는 $(M_{red}, MOD, \sigma, \pi)$를 생성한다. 이때 $\forall i \in \{1, \cdots, p\}$에 대해 $(M_{red}^i, MOD_i, \sigma_i, \pi_i) \neq (M_{red}, MOD, \sigma, \pi)$이고 $DocVerify(M_{red}, MOD, \sigma, \pi, VK_{\mathbb{F}}, PK_{SIG}, PK_{PKE}) = 1$일 경우 실험 Forge($\lambda$)는 1을 출력한다.

  따라서 공격자의 어드벤티지(advantage)를 $\mathrm{Adv}_{\mathrm{Leak}}^{A} = |\Pr[\mathrm{Forge} = 1]|$이라 정의하고, 만약 모든 확률적인 다항시간의 공격자로부터 $\mathrm{Adv}_{\mathrm{Leak}}^{A}$값이 무시할 만큼 작다면 VDRRF기법은 Unforgeability를 만족한다.

정의 1. 위에서 정의한 Privacy와 Unforgeability를 만족할 경우 VDRRF기법은 안전하다.

V. 검증 가능한 클라우드 기반의 개인 건강기록 관리 모델

본 장에서는 Ⅵ장에서 제안한 Verifiable Document Redacting with Recovery Functionality(VDRRF)기법으로 검증 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안하고, 제안하는 모델의 안전성을 증명한다.

5.1 검증 가능한 클라우드 기반의 개인 건강기록 관리 모델 정의

제안하는 모델은 다섯 개의 개체(entity)에 의해 수행된다. 첫 번째 개체는 건강기록을 발행하는 Health Record Issuer(HRI), 두 번째 개체는 개인 건강기록의 소유자인 Health Record Owner(HRO), 세 번째 개체는 소유자로부터 개인 건강기록을 외부 위탁(outsoucing)받아 검증하고 저장하여 관리하는 PHR 클라우드, 네 번째 개체는 PHR 클라우드로부터 개인 건강기록을 다운받아 사용하는 Health Record User(HRU), 마지막으로 HRO의 동의하에 HRU의 요청으로부터 개인 건강기록을 복원하는 Redact Recovery Authority(RRA)가 있다. 전체적인 흐름은 Fig. 5.과 같다.

5.1.1 개인 건강기록 발행단계

건강기록이 보편적으로 병원으로 발행되는 반면, 개인 건강기록은 병원뿐만 아니라 개인 디바이스, IoT기기등 개인에게 건강기록을 제공할 수 있는 다양한 곳으로부터 발행된다. HRI는 VDRRF기법의 *Authent*알고리즘으로 개인 건강기록에 서명을 하여 HRO에게 제공한다. 이때 서명의 무결성은 유지하며 HRO가 자신의 건강기록을 관리할 수 있도록 Redactable 서명기법[5,6]이 사용된다.

5.1.2 개인 건강기록 외부 위탁단계

HRO는 웨어러블 디바이스(wearable device)나 IoT기기로부터 실시간으로 개인 건강기록을 제공받거나 병원이나 의료기기로부터 검진 후 개인 건강기록을 제공받게 된다. 이렇게 다양하게 제공 받은 HRO는 개인 건강기록을 효율적으로 관리하기 위해 PHR 클라우드에 외부 위탁한다. HRO는 VDRRF기법의 *Redact*알고리즘을 사용하여 발생할 수 있는 사생활(privacy) 침해문제를 방지하기 위해 민감한 정보를 삭제하고, 삭제된 부분 외에는 변경 되지 않았음을 검증하기 위해 영지식 증명으로 증명(proof)을 생성하고, 자신의 개인정보가 특정 경우에 복원되어 사용할 수 있음을 정책적으로 동의하여 개인 건강기록의 삭제된 부분을 RRA의 공개키로 암호화하여 업로드한다. VDRRF기법을 사용함으로써 개인 건강기록을 암호화하여 관리하는 기법들과 달리 HRI로부터 발행되었음을 검증할 수 있고, HRO가 동의한 경우에만 민감한 정보를 공개 할 수 있음으로 민감한 정보가 삭제되었지만 암호화 되지 않은 개인 건강기록은 다양하게 사용될 수 있어 의료 데이터의 가용성을 향상 시킬 수 있다. 또한 zk-SNARK를 사용하였기 때문에 증명의 크기가 간결하며 검증시간이 빠르다는 장점이 있다.

PHR 클라우드는 HRO로부터 개인 건강기록을 외부 위탁받아 VDRRF의 *DocVerify*알고리즘으로 HRI의 서명과 HRO의 증명을 검증하고 검증을 통과할 경우 위탁 받은 개인 건강기록을 저장한다.

5.1.3 개인 건강기록 사용단계

HRU는 데이터 소비자로써 다양한 목적으로 PHR 클라우드에 저장하고 있는 개인 건강기록들에 접근한다. 예를 들어 통계데이터를 위한 통계기관이나 건강기록 확인을 위한 보험기관이 있을 수 있다. 통계기관의 목적은 민감한 정보가 삭제된 문서로 이룰 수 있으나 보험기관의 목적은 개인정보 확인이 필요하기 때문에 삭제된 문서의 복원을 필요로 한다. 예시와 같이 HRU가 삭제된 개인 건강기록의 복원이 필요할 경우 RRA에게 복원 요청을 한다.

5.1.4 개인 건강기록 복원단계

HRU로부터 복원 요청을 받은 RRA는 HRU가 동의한 정책에 해당될 경우에만 *Recovery*알고리즘을 사용하여 개인 건강기록의 삭제된 부분을 복원하여 HRU에게 전달한다.

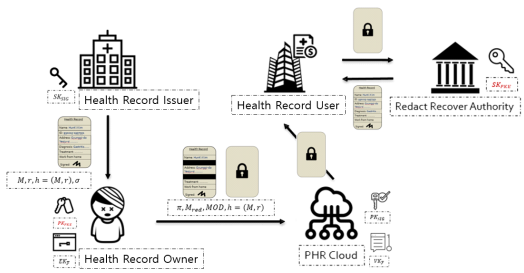

Fig. 5. Verifiable Cloud-based Personal Health Record with Recovery Functionality

5.2 검증 가능한 클라우드 기반의 개인 건강기록 관리 모델의 안전성 증명

제안하는 모델에서 PHR 클라우드에 제공하는 증명의 안전성은 일반적인 영지식 증명(zero-knowledge)의 안정성과 같이 완전성, 건정성, 영지식성을 기준으로 판단할 수 있다.

5.2.1 완전성(completeness)

제안하는 모델에서 쓰인 영지식 증명 기법은 완벽한 완전성을 가지므로 PHR 클라우드에 제공한 증명은 완전성을 가진다.

5.2.2 건전성(soundness)

제안하는 모델에서 쓰인 영지식 증명 기법은 계산적인 건전성을 가진다. 4.2장에서 제안한 기법의 안전성 정의에 의해 PHR 클라우드에 저장하기 위한 HRI의 서명은 privacy하고 위조불가능(unforgeable)하며 HRU가 삭제한 부분의 색인과 개인 건강기록으로 만들어진 암호문이 영지식 증명 생성에 이용되도록 하였으므로 악의적으로 개인 건강기록을 삭제 및 수정하는 것은 불가능하다. 따라서 PHR 클라우드에 제공한 증명은 건전성을 가진다.

5.2.3 영지식성(zero-knowledge)

제안하는 모델에서 쓰인 영지식 증명은 완벽한 영지식성을 가진다. 따라서 PHR 클라우드에 제공한 증명은 영지식성을 가진다.

VI. 구현 및 분석

6.1 구현

본 논문에서 제안한 Verifiable Document Redacting with Recovery Functionality(VDRRF)기법을 구현하기 위해 공개 소프트웨어인 libsnark[15]와 SAVER[16]를 이용하였다. libsnark는 zk-SNARK 라이브러리 중 가장 널리 쓰이며 Parno[17] 와 Groth[18]등이 제안한 검증시스템을 지원한다. 성능 측정을 위해 libsnark의 Groth의 검증시스템을 이용하였으며 이용한 보안수준 128비트의 BN 타원곡선을 사용하였다.

SAVER는 libsnark 라이브러리에 암호화 코드를 추가하여 블록체인 기반의 투표시스템을 구현한 기법이다. 암호 알고리즘을 zk-SNARK 써킷(circuit)에 구현할 경우 암호 알고리즘의 복잡한 연산 때문에 써킷의 크기가 증가하여 검증시간이 증가함으로 비효율적으로 구현이 된다. 따라서 SAVER에서는 엘가말(ElGamal) 기반의 검증 가능한 암호기법으로 메시지를 암호화하여 SNARK 써킷과 결합하고 이를 활용해 투표시스템을 구현하였다. SAVER의 암호 코드를 이용하여 VDRRF기법에서 Redact Recovery Authority(RRA)를 통해 복호화 되는 암호문을 만드는 알고리즘을 구현하였다.

6.2 분석

본 논문에서 제안한 VDRRF 기법은 Ajtai 해쉬함수[19]로 기법의 *Authent*알고리즘에서 사용된 해쉬써킷을 구현해 해쉬값에 대한 영지식 증명을 하였으며 기법의 *Redact*알고리즘의 메시지 삭제과정을 *Redact*써킷으로 구현해 색인번호(index number)에 해당하는 메시지를 삭제하고 색인번호 외에는 변경되지 않았음에 대한 영지식 증명을 하였다. 또한 SAVER[15]를 이용해 기법의 *Enc*알고리즘을 구현해 메시지를 암호화 하여 RRA로부터 복원 받은 메시지가 원본메시지임에 대한 영지식 증명을 하였다. 이때 SAVER에서 사용된 엘가말 기반의 암호화 알

Table 2. Verifiable Document Redacting with Recovery Functionality performance

Message	Constraints	EK size	VK size	Proving time	Verifying time
m=128 bit	383	106.4 KB	0.15 KB	0.015 sec	0.01 sec
m=256 bit	511	163.1 KB	0.15 KB	0.024 sec	0.01 sec
m=512 bit	767	276.7 KB	0.15 KB	0.043 sec	0.01 sec
m=1024 bit	1279	503.7 KB	0.15 KB	0.076 sec	0.01 sec

고리즘은 지수값을 메시지로 다루어 암호화하기 때문에 메시지는 이산대수가 풀리는 범위 내에서 작은 메시지로 선택해야 한다. 모든 프로그램은 C++를 이용하여 구현하였고, Intel Core i5-4590 3.30GHz RAM 16GB 환경에서 실험하였다.

각각 128, 256, 512, 1024 비트의 메시지로 제안 기법의 constraints의 개수, 증명키와 검증키의 크기, 증명 시간, 검증 시간에 대해 측정하였다. 각각의 메시지 크기별로 측정한 결과 값은 다음 Table 2.와 같다.

메시지의 크기가 증가하면서 검증키의 크기는 일정정하며 검증시간은 증가하지만 크게 차이가 나지 않는다. 하지만 constraints의 개수는 메시지 크기에 따라 일정하게 증가하고 증명키의 크기는 크게 증가하며 증명시간 또한 증가하는 것을 알 수 있다. 또한 증명키와 검증키의 크기 차이가 크지만 증명시간과 검증시간의 차이는 크지 않다는 것을 알 수 있다.

따라서 증명시간과 검증시간이 빠르며 증명키의 크기가 크지만 다수의 개인 건강기록을 저장하는 클라우드의 검증키는 작기 때문에 클라우드 기반의 개인 건강기록 관리모델에 사용 될 수 있음을 확인하였다.

VII. 결 론

본 논문에서는 영지식 증명과 Redactable 서명 기법을 사용함으로써 메시지의 민감한 정보를 삭제하해 프라이버시를 보호하며 Redact Recovery Authority를 통해 복원할 수 있는 기법을 제안하였다. 또한 제안하는 기법을 활용해 검증 가능한 클라우드 기반의 개인 건강기록 관리모델을 제안하였다. 마지막으로 제안한 기법을 설계한 후 실험하여 기법의 성능을 분석하였다.

개인 건강기록은 자신의 건강기록을 관리한다는 개념으로 시작되었으나 효율적인 관리를 위해 클라우드에 위탁되어 관리하고 있다. 제안하는 기법을 사용함으로써 데이터 소유자는 개인 건강기록의 민감한 정보를 선택적으로 삭제하고 외부 위탁하기 때문에 민감한 정보에 대한 주체가 되는 장점이 있다. 또한 클라우드를 통해 의료 데이터를 사용하는 데이터 소비자는 민감한 정보가 삭제된 데이터를 사용하거나 필요에 의해 복원 받아 사용할 수 있기 때문에 선택적인 접근으로 의료 데이터의 가용성이 증가되는 장점이 있다. 다만 소유자를 거치지 않고 Redact Recovery Authority를 통해 복원이 이루어지기 때문에 명확하고 강화된 정책에 대한 소유자의 동의가 필수적이어야 한다.

References

[1] Tang, Paul C., et al, "Personal health records: definitions, benefits, and strategies for overcoming barriers to adoption," *Journal of the American Medical Informatics Association,* 13.2, pp. 121-126, 2006

[2] SM. Li, S. Yu, N. Cao, W. Lou, "Authorized private keyword search over encrypted data in cloud computing," *2011 31st International Conference on Distributed Computing Systems,* pp. 383-392, 2011

[3] Goldwasser, Shafi, Silvio Micali, and Charles Rackoff, "The knowledge complexity of interactive proof systems," *SIAM Journal on computing*, 18.1, pp. 186-208, 1989

[4] Gennaro, Rosario, Craig Gentry, and Bryan Parno, "Non-interactive verifiable computing: Outsourcing computation to untrusted workers," *In Annual Cryptology Conference*, pp. 465-482, 2010

[5] Johnson, Robert, et al, "Homomorphic signature schemes," *Cryptographers' track at the RSA conference,* pp. 244-262, 2002

[6] Slamanig, Daniel, and Stefan Rass, "Generalizations and extensions of redactable signatures with applications to electronic healthcare," *IFIP International Conference on Communications and Multimedia Security*, pp. 201-213, 2010

[7] Chabanne, Hervé, Rodolphe Hugel, and Julien Keuffer, "Verifiable document redacting," *European Symposium on Research in Computer*

Security, pp. 334-351, 2017

〔8〕 Bethencourt, John, Amit Sahai, and Brent Waters, "Ciphertext-policy attribute-based encryption," *2007 IEEE symposium on security and privacy (SP'07)*, pp. 321-334, 2007

〔9〕 Wang, Changji, Xuan Liu, and Wentao Li, "Design and implementation of a secure cloud-based personal health record system using," *International Journal of Intelligent Information and Database Systems 4*, 7(5), pp.389-399, 2013

〔10〕 Gondkar, Deepali A., and V. S. Kadam, "Attribute based encryption for securing personal health record on cloud," *2014 2nd International Conference on Devices, Circuits and Systems (ICDCS)*, pp. 1-5, 2014

〔11〕 Rao, Y. Sreenivasa, "A secure and efficient ciphertext-policy attribute-based signcryption for personal health records sharing in cloud computing," *Future Generation Computer Systems*, 67, pp. 133-151, 2017

〔12〕 Deng, Fuhu, et al, "Ciphertext-policy attribute-based signcryption with verifiable outsourced designcryption for sharing personal health records," *IEEE Access 6*, pp. 39473-39486, 2018

〔13〕 N. Bitansky, A. Chiesa, Y. Ishai, O. Paneth, and R. Ostrovsky, "Succinct non-interactive arguments via linear interactive proofs," *Theory of Cryptography Conference*, pp. 315-333, 2013

〔14〕 Naveh, Assa, and Eran Tromer, "PhotoProof: Cryptographic image authentication for any set of permissible transformations," *2016 IEEE Symposium on Security and Privacy (SP)*, pp. 255-271, 2016

〔15〕 GitHub, "libsnark: a C++ library for zkSNARK proofs," https://github.com/scipr-lab/libsnark, Oct. 2019

〔16〕 GitHub, "SAVER," https://github.com/snp-lab/SAVER, 2019

〔17〕 Parno, Bryan, et al, "Pinocchio: Nearly practical verifiable computation," *2013 IEEE Symposium on Security and Privacy*, pp. 238-252, 2013.

〔18〕 J. Groth, "On the size of pairing-based non-interactive arguments," *EUROCRYPT 2016*, pp. 305-326, May. 2016

〔19〕 M. Ajtai, "Generating hard instances of lattice problems (extended abstract)," *ACM Symposium on Theory of Computing*, pp. 99-108, Jul. 1996

〈저 자 소 개〉

김 훈 기 (Hunki Kim) 학생회원
2017년 8월: 경희대학교 응용수학과/컴퓨터공학과 졸업
2017년 9월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 암호 프로토콜, 암호이론, 영지식 증명, 블록체인 기반의 암호기술

김 종 현 (Jonghyun Kim) 학생회원
2014년 2월: 성균관대학교 수학과 졸업
2014년 3월~현재:고려대학교 정보보호대학원 석박사 통합과정
〈관심분야〉 암호 프로토콜, 암호이론, 함수 암호

이 동 훈 (Dong Hoon Lee) 종신회원
1983년 8월: 고려대학교 경제학과 졸업
1987년 12월: Oklahoma University 전산학과 석사 졸업
1992년 5월: Oklahoma University 전산학과 박사 졸업
1993년 3월~1997년 2월: 고려대학교 전산학과 조교수
1997년 3월~2001년 2월: 고려대학교 전산학과 부교수
2001년 3월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 암호 프로토콜, 암호이론, USN이론, 키 교환, 익명성 연구, PET 기술