클라우드 컴퓨팅에서 u-Fitness 기반 운동 관리를 위한 소유자의 프라이버시를 보장하는 프로토콜

(A Owner's Privacy Preserving Protocol for u-Fitness-based Exercise Management in Cloud Computing)

김태연*, 조기환**, 최은복***

(Tae-yeon Kim, Ki-hwan Cho, Eun-Bok Choi)

요 약

u-fitness 기반 운동관리와 관련된 민감한 신체 정보의 양이 폭발적으로 증가하고 있기 때문에 클라우드 서비스의 활용에 대한 관심이 날로 증가하고 있다. 그러나 클라우드 서버는 자신의 서버에 저장된 정보에 불법적으로 접근할 수 있으며, 누가 그 정보의 소유자인지를 알아낼 수 있고, 스토리지에 저장된 정보 간의 연관성을 불법적으로 추론할 수 있다. 또한 클라우드 서버는 저장된 정보에 대해 수정, 삭제 같은 소유자의 정당한 연산 요청에 대해 불이행할 수 있으며, 자체의 오작동으로 인해 정보를 분실하거나 손상시킬 수 있다. 따라서 우리는 클라우드 서버를 전적으로 신뢰할 수 없기 때문에 클라우드 서버를 사용하기 위해서는 위와 같은 문제들을 해결해야 한다. 우리는 클라우드 컴퓨팅 환경에서 u-Fitness 기반 운동관리를 위한 소유자의 프라이버시를 보장하는 프로토콜을 제안한다. 그리고 제안한 구조가 실제 환경에서 적용 가능함을 보안 분석과 성능 분석을 통해 보인다.

■ 중심어 : u-휘트니스 ; 프라이버시 ; 운동관리 ; 클라우드 컴퓨팅 ; 에이전트

Abstract

There is growing interest in the use of cloud services these days because the amount of sensitive physical information related to u-fitness-based exercise management increase in explosive. However, it is possible to illegally access information stored in a cloud server, and to find out who owns the information, even, to illegally deduce an association among the information stored in its memory. The cloud server may also intentionally pass over the owner's legitimate operation requests such as modification and deletion of stored information, and may lose or damage information due to its malfunction. So, it is strongly required to solve the above problems because we can not trust the cloud server entirely. In this paper, we propose a protocol to preserve the privacy of the owner for u-Fitness-based exercise management in a cloud computing environment. And we show that our proposed architecture is applicable in real environment through security analysis and performance analysis.

■ keywords : u-Fitness ; privacy ; exercise management ; cloud computing ; agent

Ⅰ. 서 론

인터넷과 스마트환경에서 다양성, 신속성을 바탕으로 자기 투자형 생활건강관리 서비스에 대한 수요자 요구가 확산되고 있고, 그에 따른 웰니스(welless) 산업이 발전하고 있다[1,2,3]. 개인 사용자가 자신의 기본적인 개인정보와 생체 측정 기기에서 얻어진 생체 정보(체지방, 심박 수 등)를 u-Fitness 센터로 전송하면, 운동 전문가나 의사는 이러한 정보를 기반으로 현재 상태에 가장 적합한 운동 처방 프로그램을 센터 서버 스토리지에 저장한다. 그러면 센터는 모바일 디바이스를 통해 그 처방 프로그램을 사용자에게 전송하는 등 개인 사용자에게 최적화된 맞춤형 서비스를 제공한다[4,5].

이러한 신체 정보는 시간이 지남에 따라 그 양이 폭발적으로 증가하는데, 현재 대부분의 서비스 제공 업체는 이러한 정보를 저장할만한 충분한 스토로지를 갖추고 있지 못할 뿐만 아니라 보안 서비스도 제대로 제공하지 못하고 있다. 이러한 문제점을 해결하기 위해 IT 인프라의 이용성과 확장성, 비용 측면에서 많

* 정회원, 서남대학교 컴퓨터정보학과
** 종신회원, 전북대학교 컴퓨터공학부
*** 정회원, 전주대학교 스마트미디어학과

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었음 (IITP-2017-2015-0-00378).

접수일자 : 2017년 08월 25일
수정일자 : 2017년 09월 18일
게재확정일 : 2017년 09월 25일
교신저자 : 조기환 e-mail : ghcho@chonbuk.ac.kr

은 장점이 있는 클라우드 컴퓨팅이 해결방안으로 대두되고 있다[6].

클라우드 컴퓨팅 환경에서 사용자는 필요한 IT 자원을 물리적으로 소유하지 않는 상태에서 시간적인 제약을 받지 않고 원격으로 접속해서 필요한 만큼의 자원을 사용할 수 있다[7]. 정보 소유자는 일정한 간격으로 수집된 정보를 클라우드 서버로 아웃소스하고, 정보 사용자는 클라우드 서버에 저장된 데이터를 필요할 때마다 다운로드하는 구조이기 때문에 클라우드 서버에 저장되는 정보의 프라이버시를 보호하는 메커니즘이 반드시 필요하다.

다중 정보 소유자와 다중 정보 사용자, 그리고 전적으로 신뢰성이 보장되지 않는 서버들로 구성되는 클라우드 환경에서 데이터의 프라이버시를 보호하기 위해 최근 정보 암호화나 키워드 암호화, 효율적인 검색을 위한 구조들이 제안되고 있지만 다음과 같은 신뢰할 수 없는 서버를 통한 불법적인 행위들의 해결책이 구체적으로 제시하지 않아 많은 문제점을 내포하고 있는 실정이다.

첫째로, 서버에 아웃소스하는 정보 소유자의 신원과 서버에 저장된 정보를 다운로드하는 사용자의 신원을 알아내는 행위이다. 두 번째는 서버가 자신의 스토리지에 저장된 정보 검색을 요청하는 질의를 기반으로 해서 불법적으로 정보들 간의 연관성을 추론하는 행위이다. 마지막으로 자신의 스토리지에 저장된 정보에 대한 소유자의 수정과 삭제와 같은 요청 행위에대한 불이행이나 자체의 오작동으로 인한 과실(분실, 손상 등)이다.

이러한 문제점을 해결하기 위해 본 논문에서는 정보 소유자(또는 정보 사용자)들과 클라우드 서버 사이에 신뢰받는 에이전트(Trusted Agent)를 둔다. 그리고 정보 소유자(또는 정보사용자)들은 신뢰받는 에이전트와의 통신을 통해 메시지를 교환함으로써 정보 프라이버시의 침해를 최소화한다. 그리고 정보들을 암호화하는데 이용되는 비밀키들을 효율적으로 관리하기 위해 혼합 해시 체인을 사용하고, 스토리지에 저장된 정보들 간의 불법적인 연관성 추론 문제를 최소화하였다. 또한, 서버로 아웃소스된 정보들을 정기적 또는 비정기적으로 감사(auditing)하는 메커니즘에 대한 보안과 성능분석을 통하여 제안한 u-Fitness 기반 운동관리 시스템이 실제 클라우드 컴퓨팅 환경에서 적용 가능함을 증명하였다.

본 논문의 구성은 다음과 같다. 1장 서론에 이어, 2장은 관련 연구와 본 논문을 이해하는데 필요한 기본적인 용어를 정의한다. 3장은 본 논문에서 제안하는 시스템의 구조에 대해 기술하며, 4장은 본 시스템의 검색 및 감사 프로토콜에 대해 서술한다. 5장은 보안 및 성능 분석을, 그리고 마지막 6장에서 결론을 맺는다.

Ⅱ. 배 경

1. 관련 연구

건강관리 분야의 클라우드 컴퓨팅에 대한 관심이 증가하고 있지만 성공적인 구현 사례는 아직까지 거의 없으며 많은 논문에서는 클라우드 패러다임의 이점에 대한 기술이 없이 '클라우드'를 가상 머신의 사용 또는 웹 기반이라는 용어와 동의어로 사용하고 있는 실정이다[8]. 대부분의 연구들은 건강관리를 위해 다양한 IoT 제품을 사용하여 신체 정보를 수집하는 기초단계에 많은 관심을 가지고 있을 뿐 그곳에서 나온 데이터 관리에 대한 효율적인 방법은 제시되고 있지 않고 있다. 국내의 관련된 연구로는 신성훈[1]은 웰니스 분야의 ICT 융합 기술 동향을 생활건강관리, 피트니스/건강관리, 웰에이징 분야로 구분하여 분석하였다. Griebel[8] 등은 데이터 관리와 같은 전통적인 분야가 아닌 의료 분야에서 클라우드 컴퓨팅에 대한 연구에서 현재 상태와 미래의 전망을 기술하는 등 클라우드 환경에서 구체적인 모델이나 구현은 찾아볼 수 없는 실정이다. 박성빈 등[4]은 실험실 체력평가 및 기초체력평가를 토대로 DB를 구축하고 이를 기반으로 개인의 체력상태를 반영한 건강관리 서비스를 제공하는 u-fitness 기반의 맞춤형 운동 관리 시스템을 개발하였다. 이는 수집한 데이터를 전달하는 과정이나 관리하는 과정에서 발생할 수 있는 데이터 프라이버시의 보호에 대해서는 언급이 없다. 반면에 Jeong[9]은 사용자의 헬스케어 정보를 안전하게 저장하고 처리, 관리될 수 있도록 헬스케어시스템의 센서 정보에 속성 값을 배정하여 사용자의 프라이버시를 통합 관리하는 계층적 구조를 제안하였다. 하지만 이 구조는 소규모의 환경에서 전적으로 서버를 신뢰한다는 가정 하에서 접근제어 정책을 기반으로 하여 구현되었고 민감한 데이터의 비밀성과 연관성 추론등과 보안 기능에 대한 언급이 없다.

Chen 등[10]은 암호화된 클라우드 데이터에 대해 의미론적 멀티 키워드 기반과 순위가 고려된 파일 검색이 가능한 방식을 제안하였다. 이 방식은 검색 키워드와 정확히 일치하는 파일뿐만 아니라 의미론적으로 키워드와 관련된 용어를 포함하는 파일도 다운로드할 수 있도록 하였다.

Boneh 등[11]은 공개키 기반에서 순위가 고려된 검색 구조를 제안하였다. 그들의 구조는 쌍 선형 사상과 트랩도어 순열을 사용하기 때문에 처리 시간이 많이 걸리는 단점이 있었다. 그리고 Zhang 등[12]은 다수의 데이터 소유자가 연결된 클라우드 환경에서 데이터의 프라이버시와 사용자의 효율성을 보호할 수 있는 순위가 고려된 멀티 키워드 검색 프로토콜을 제안하였다. 사용자의 요청 질의 내의 암호화된 키워드가 인덱스 테이블 내의 키워드와 서로 일치하는 것이 있는지를 판단하기 위해 쌍 선형 페어링 연산을 수행하고, 특수 함수를 사용하여 키워드들과 관련된 데이터 파일들 간에 관련성 점수을 계산한 다음에, 관련성

점수가 높은 k개의 데이터를 다운로드하도록 하는 방식이다. 앞에서 기술한 Chen 등[10]과 Boneh 등[11], Zhang 등[12]이 제안한 구조들은 아웃소스된 다수의 소유자의 데이터들에 동일한 키워드가 있다는 조건에서 데이터의 소유자가 누구든 관계없이 안전하고 효율적으로 원하는 데이터를 검색하는 메커니즘을 사용하고 있기 때문에 개인의 건강관리 시스템 환경에서는 적합하지 않다.

따라서 본 논문에서는 멀티 정보블록 식별자를 기반으로한 신뢰받는 에이전트 아키텍쳐를 통해 한 특정 소유자의 데이터를 안전하고 효율적으로 검색할 수 있는 소유자의 프라이버시 보장 프로토콜을 제안하였다.

2. 용어 정의

본 논문에서 사용하는 용어들을 정리하면 아래와 같다.

- i, i' : 클라이언트(소유자와 사용자) 식별자의 인덱스
- j, j' : 블록 식별자의 인덱스
- s, s' : 데이터 블록 수
- $m_{i,j}$: 클라이언트 i의 j번째 블록
- $H(\bullet)$: 일방향 해시 함수
- $BI_{i,j}(=H(m_{i,j}))$: 클라이언트 i의 j번째 블록 식별자
- U : 전체 클라이언트들의 식별자 집합
- UI : 전체 데이터 블록들의 식별자 집합
- UI_i : 소유자 i의 데이터 블록들의 식별자 집합
- $UI_i^{'}\ (= \bigcup_{j\in\{1,2,\ldots,s\}}^{s'} H(m_{i,j}))$: 소유자 i의 블록 식별자들 중에서 s'개의 식별자로 구성된 집합, $|UI_i^{'}| = s'$, $UI_i^{'} \subset UI_i$
- $UI^{'}(= \bigcup_{i\in U, j\in\{1,2,\ldots,s\}}^{s'} H(m_{i,j}))$: 전체 블록 식별자들 중에서 s'개의 식별자로 구성된 집합, $|UI^{'}| = s'$, $UI^{'} \subset UI$

Ⅲ. 시스템 모델

1. 시스템 구조

정보소유자들은 클라우드 서버에게 자신의 파일들을 안전하게 아웃소스하고, 정보사용자들은 서버에 저장된 파일들을 효율적으로 접근 할 수 있어야 한다. 다시 말해서 정보소유자는 파일을 서버에게 아웃소스하기 전에 암호화해서 전송하고, 사용자는 서버로부터 다운로드한 파일을 소유자가 제공한 비밀키를 사용하여 복호화한 후에 접근하는 메커니즘이 필요하다. 우리가 제안한 시스템 구조는 그림 1과 같이 다중 소유자(O : $Owner$)와 다중 사용자(Ur : $User$), 신뢰받는 에이전트(TA : $Trusted\ Agent$), 클라우드 서버(CS: $Cloud\,Server$)로 구성되며, 멀티-정보블록 식별자를 기반으로 안전한 데이터 검색을 한다.

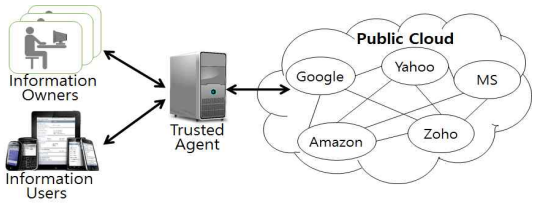

그림 1. 제안한 클라우드 컴퓨팅 시스템 구조

서버 CS는 소유자의 요청에 의해 정보를 저장하고 관리하는 노드이고, 소유자 O는 수집된 정보를 CS로 아웃소스하거나 필요에 따라 다운로드하여 접근하는 노드, 사용자 Ur(의사나 운동 트레이너)는 CS에 저장된 소유자들의 정보를 다운로드하여 접근하거나 관련된 정보를 O에게 전달하는 노드, TA는 클라이언트(O 또는 Ur)와 CS 사이에 존재하며 클라이언트의 에이전트 역할을 수행하는 노드이다.

2. 파일 식별자 생성

각 소유자 O_i는 수집된 정보를 블록 단위로 관리한다. O_i의 정보가 $s(\geq 1)$개의 블록($m_{i,1}, m_{i,2}, \cdots, m_{i,s}$)으로 분할된 경우, 일방향 해시함수의 결과 값인 $H(m_{i,j})$을 블록 $m_{i,j}$의 식별자로 사용한다. 식별자 기반 인덱스는 정보 블록 식별자와 $ACL(Access\ Control\ List)$, 임의의 난수($r_{i,j}$), 무결성 검사값($V_{i,j}=H\left(r_{i,j} \| [m_{i,j}]_{K_{O_i}}\right)$)으로 구성된다. 여기에서 ACL는 블록 $m_{i,j}$을 접근할 수 있는 클라이언트들의 식별자 리스트이고, $V_{i,j}$는 블록 $m_{i,j}$의 무결성을 체크하는데 사용된다.

3. 비밀키 생성과 관리

소유자 O_i의 정보는 시간이 지남에 따라 증가하기 때문에 소유자 O_i가 관리해야 할 정보 블록의 수도 늘어나고 그만큼의 비밀키도 관리해야 한다. 이러한 문제를 최소화하기 위해 그림 2와 같이 전방향(forward) 해시 체인 값과 후방향(backward) 해시 체인 값을 혼합 한 해시 체인을 값을 비밀키로 사용하는 메커니즘을 사용한다[13].

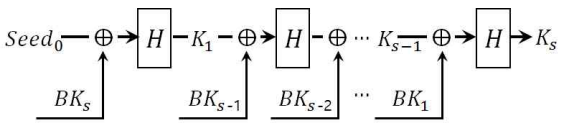

그림 2. 혼합 해시 체인

예를 들어, 첫 번째 비밀키는 $H(Seed_O \oplus BK_s)$에 의해서 생성되고 그 다음 비밀키들은 $H(K_1 \oplus BK_{s-1})$와 $H(K_2 \oplus BK_{s-2})$, ..., $H(K_{s-1} \oplus BK_1) = K_s$순으로 생성된다. 여기에서 $K_j(j=1,...,s)$는 전방향 해시 체인이고, $BK_j(j=1,...,s)$는 후방향 해시 체인이다.

만약 공격자가 전방향 해시 체인 값을 안다고 하더라도 후방향 해시 체인을 모르는 상태에서 새로운 비빌키를 생성하는 것은 매우 어렵다. 이러한 구조를 사용하는 목적은 두 해시 함수의 초기 $Seed_0$ 값만 안전하게 관리하면 비밀키를 간단하고 효율적으로 관리할 수 있기 때문이다.

Ⅳ. 정보검색과 감사 프로토콜

1. 정보 검색

가. 준비 단계

1) 준비 단계

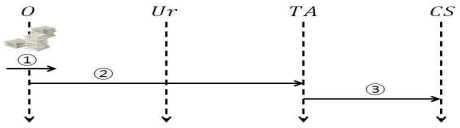

그림 3. 준비단계

[절차1a] $O_i \rightarrow TA$:

$$\left[H(m_{i,j}), [m_{i,j}]_{K_{O_i}}, ACL_{i,j}\right]_{K_{O_i,TA}}, \quad \forall i \in U,\ 1 \le j \le s$$

절차 1a는 수집된 정보를 서버로 아웃소스하기 위한 첫 번째 절차로서 소유자 O_i는 자신의 j번째 정보 블록 식별자와 자신의 비밀키로 암호화한 정보 블록, 그리고 j번째 정보 블록의 접근권한을 나타내는 ACL을 TA와의 세션키로 암호화한 다음에 CS가 아닌 신뢰받는 에이전트 TA에게 전송한다(그림 3의 ②). 그리고 O_i는 비밀키를 생성하는데 사용될 사용될 2개의 입력 값($Seed_i, Seed_i'$)을 관리하고, 차후에 TA가 정상적인 임무완료 후 전송된 정보 블록은 삭제한다. 여기에서 $m_{i,j}$는 O_i의 j번째 정보 블록이고 $H(m_{i,j})$는 정보 블록의 식별자, $ACL_{i,j}$는 j번째 정보 블록에 대한 접근제어 리스트, $K_{O_i,TA}$는 O_i와 TA 간에 사용될 비밀키이다. 그리고 $Seed_i$와 $Seed_i'$는 각각 전방향 해시 체인과 후방향 해시 체인을 생성하는데 사용되는 초기 값이다(그림 2 참조).

[절차 1b] $O_i \rightarrow TA$: $\left[-, H(m_{i,j})\right]_{K_{O_i,TA}}$

절차 1b는 O_i가 CS로 아웃소스한 블록 $m_{i,j}$의 삭제를 요청하는 메시지이다. 여기에서 '−'은 '블록 삭제'를 나타낸다.

[절차 2a] $TA \rightarrow CS$: $\left[H(m_{i,j}), [m_{i,j}]_{K_O}\right]_{K_{TA,CS}}$

절차 2a는 TA가 O_i로부터 아웃소스 요청 메시지를 수신하면 복호화한 다음에 블록 식별자 인덱스에 (정보 블록 식별자, $ACL_{i,j}$, 난수($r_{i,j}$), 무결성 검사 값($V_{i,j}$))를 추가하고, 정보 블록의 식별자와 암호화된 정보 블록을 CS와의 세션키로 암호화한 후에 CS에게 전송한다.

[절차 2b] $TA \rightarrow CS$: $\left[-, H(m_{i,j})\right]_{K_{TA,CS}}$

절차 2b는 O_i의 '삭제 요청' 메시지를 수신한 TA가 CS에게 삭제요청을 중계하는 역할을 수행하는 것이다. O_i의 삭제 요청은 특정 블록을 수정하거나 블록이 더 이상 의미가 없어진 경우에 수행한다. 이 중에서 블록을 수정하는 과정은 (a) 블록의 다운로드→삭제 요청→수정→아웃소싱, (b) 블록의 다운로드→수정→삭제 요청→아웃소싱, (c) 블록의 다운로드→수정→아웃소싱 →삭제 요청 등이 있다. 따라서 TA는 다운로드한 후에 해당 블록을 반드시 삭제해야 하지만 CS가 다운로드한 블록과 삭제 요청 블록, 아웃소스한 블록들 간의 연관성을 추론할 가능성을 최소화하기 위해 다운로드나 아웃소스 요청을 수행한 후 일정 시간이 경과한 후에 삭제 요청을 한다.

나. 검색 단계

아웃소스된 정보가 클라우드 서버에서 안전하게 관리되는 것도 중요하지만 효율적인 검색이 되도록 하는 것도 이에 못지않게 중요하다. CS에 저장되어 있는 정보 블록을 검색하는 주체는 O와 Ur로서 검색하는 절차(그림 4, 그림 5 참조)와 형식이 두 주체들 간에 차이가 있다.

O인 경우 정보를 복호화하는데 사용되는 비밀키를 자신이 관리하기 때문에 자신이 아웃소스한 정보를 다운로드해서 복호화한 다음 바로 읽을 수 있지만 Ur인 경우에는 비밀키를 가지고 있지 않기 때문에 O로부터 비밀키를 수신하는 절차가 필요하다. 그림 4는 정보소유자인 O의 측면에서 정보를 검색하기 위해 TA를 거쳐 CS로부터 해당 정보 블록을 다운로드하는 과정을 나타낸 것이고, 그림 5는 정보 사용자인 Ur의 측면에서 정보를 검색하기 위해 정보소유자로부터 비밀키를 수신한 다음에 원하는 해당 정보 블록을 CS로부터 다운로드하는 과정을 나타낸 것이다.

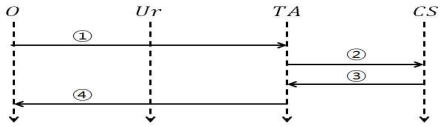

그림 4. 정보소유자 측면

[절차 3a] $O_i \rightarrow TA$: $[\bigcup_{j' \in \{1,2,...,s\}}^{s'} H(m_{i,j'}), k]_{K_{O_i TA}}$,

$|UI_i'| = s'$, $UI_i' \subset UI_i$

절차 3a는 O_i가 정보 요청 메시지를 직접 CS에게 전송하지 않고 신뢰받는 에이전트인 TA에게 전송하는 과정을 나타낸 것이다. 메시지를 TA로 전송하는 이유는 요청하는 정보 블록들의 소유자나 그 사용자의 신원을 CS에게 노출시키지 않기 위함이다. 다시 말해서 TA로 하여금 에이전트 역할을 하도록 함으로써 정보소유자나 정보사용자의 신원을 감출 수 있기 때문이다.

여기에서 $UI_i' = \bigcup_{j' = \{1,2,...,s\}}^{s'} H(m_{i,j'})$는 O_i가 자신이 아웃소스한 블록들 중에서 검색을 원하는 블록들의 식별자들을 나타낸 것으로 k개를 넘을 수 없다. 그리고 변수 k는 보안 강도를 지정하는 것으로서 TA로 하여금 k개의 블록 식별자들로 구성된 트랩도어를 생성하라는 의미이다(절차 4 참조). 따라서 O_i가 요청한 식별자들의 수와 지정한 k의 값의 차이가 클수록 보안 강도는 높다.

정보사용자 측면에서 클라우드 서버에 저장되어 있는 정보를 다운로드하기 위해서는 절차 3b-1과 절차 3b-2, 그리고 절차 3b-3과 같이 CS에게 메시지를 전송해야한다.

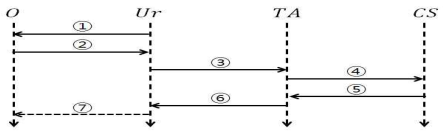

그림 5. 정보사용자 측면

다시 말해서 Ur는 아웃소스된 정보를 다운로드할 때마다 원하는 블록들의 식별자와 복호화 키들을 O_i로부터 수신한 후에 절차 3b-2과 절차 3b-3의 과정을 통해 TA를 거쳐 CS에게 정보를 요청한다.

[절차 3b-1)] $Ur_i \rightarrow O_{i'}$:

$$\left[time, \bigcup_{j=\{1,2,...,s\}}^{s'} H(m_{i',j})\right]_{K_{Ur_i O_i}},$$

$|UI_i'| = s'$, $UI_i' \subset UI_i$

절차 3b-1는 Ur가 $time$와 검색을 원하는 s'개의 블록 식별자들을 O_i에게 전송하는 과정이다. 여기에서 $time$은 재전송을 방지하기 위한 타임스탬프이다.

[절차 3b-2] $O_{i'} \rightarrow Ur_i$:

$$\left[[\bigcup_{j \in \{1,2,...,s\}}^{s'} H(m_{i',j}), k]_{O_{i'}, TA}, Seed_{i'}, Seed'_{i'}\right]_{K_{O_{i'} Ur_i}},$$

$|UI_i'| = s'$, $UI_i' \subset UI_i$

[절차 3b-3] $Ur_i \rightarrow TA$: $[\bigcup_{j \in \{1,2,...,s\}}^{s'} H(m_{i',j}), k]_{O_{i'}, TA}$,

$|UI_i'| = s'$, $UI_i' \subset UI_i$

절차 3b-2는 Ur_i가 O_i로부터 암호화된 블록 식별자들과 해시 체인을 생성하는데 사용되는 2개의 초기값을 받는 과정이다. 여기에서 $Seed_{i'}$와 $Seed_{i'}'$는 각각 전방향 해시 체인과 후방향 해시 체인을 위한 것으로 초기 값들이 아닌 Ur_i에게 접근 권한이 있는 정보로 부터 복호화할 수 있는 값들이다. 절차 3b-3은 O_i로부터 수신한 메시지를 복호화한 다음에 암호화된 식별자들의 리스트만 TA에게 전송하는 과정이다.

TA는 수신한 메시지를 복호화한 다음에 블록 $\bigcup_{j' \in \{1,2,...,s\}}^{s'} H(m_{i,j'})$들을 O_i나 $Ur_{i'}$가 접근권한이 있는지를 ACL을 검색하여 검사한다. 접근권한이 없는 경우는 무시하고, 그렇지 않는 경우에는 절차 4와 같이 트랩도어를 생성하여 CS에게 전송한다.

[절차 4] $TA \rightarrow CS$: $UI' = [\bigcup_{\exists i' \in U, j \in \{1,2,...,s\}}^{k} H(m_{i',j})]_{K_{TA,CS}}$,

$|UI'| = k$, $UI' \subset UI$

TA는 k개의 식별자로 구성된 트랩도어를 생성하여 절차 4와 같이 CS에게 전송한다. 트랩도어 내의 블록 식별자들은 O_i가 지정한 식별자들과 식별자 인덱스에서 중복이 없이 임의로 선택한 식별자들이다. 그리고 CS가 해당 블록들 간의 연관성을 추론하는 것을 최소화하기 위해 트랩도어 내의 식별자들을 나열하는 순서는 일정한 규칙이 아닌 임의적으로 한다.

[절차 5] $CS \rightarrow TA$: $UI' = \left[\bigcup_{\exists i \in U, j \in \{1,2,...,s\}}^{k} [m_{i,j}]_{K_{O_i}}\right]_{K_{TA,CS}}$,

$|UI'| = k$, $UI' \subset UI$

[절차 6] $TA \rightarrow O_i$ or Ur_i:

$$\left[\bigcup_{j \in \{1,2,...,s\}}^{s'} [m_{i,j}]_{K_{O_i}}\right]_{K_{O_i TA}} \text{ or } \left[\bigcup_{j \in \{1,2,...,s\}}^{s'} [m_{i,j}]_{K_{O_i}}\right]_{K_{Ur_i TA}},$$

$|UI'| = s'$, $UI' \subset UI$

절차 5는 k개의 암호화된 블록들을 TA에게 전송하는 과정이고, 절차 6은 CS로부터 수신한k개의 암호화된 블록들 중에서 O_i(또는 Ur_i)가 요청한 s'개의 식별자들의 암호화된 블록들만 필터링하여 전송하는 과정이다. 메시지를 수신한 O_i(또는 Ur_i)은 블록들을 복호화한 다음에 접근한다.

2. 감사 기능

클라우드 환경에서 보안 취약점은 악의적인 공격자나 서버가 원격지에 저장된 정보를 불법적으로 수정하고 삭제하는 행위, 사용자 정보가 삭제된 사실을 통보하지 않는 행위 그리고 정당한 사용자가 요청한 정보 삭제 연산을 실행하지 않는 행위 등이 있다. 따라서 정기적 또는 비정기적으로 원격지 서버에 저장된 정보가 안전하게 관리되고 있는지를 감사하는 시스템이 필요하다. 이러한 무결성 감사는 모바일 기기의 특성을 고려하여 정보의 비밀성은 최대한 보장되면서 클라이언트의 관여를 최소화하기 위해 이 절에서는 에이전트인 TA가 감사 기능을 수행하는 과정을 기술한다.

[절차 7] $TA \rightarrow CS$: $[\bigcup_{\exists i' \in U, j \in \{1,2,\ldots s\}}^{s'} \{H(m_{i',j}), r_{i',j}\}]_{TA,CS}$,

$\forall i' \in U,\ 1 \le j \le s$

TA는 자신이 관리하고 있는 식별자 인덱스에서 임의로 선택된 s'개의 $\{H(m_{i',j}),\ r_{i',j}\}$을 비밀키로 암호화한 다음에 CS에게 전송한다. 여기에서 $r_{i',j}$는 해당 블록의 무결성을 감사하는데 사용되는 난수이다.

[절차 8] $CS \rightarrow TA$: $[Veri_{CS}, [m_{i',j}]_{K_{O_i}}]_{TA,CS}$

CS는 해시 함수의 출력 값인 V_rand와 서버에 저장된 블록을 TA에게 전송한다. 여기에서 $Veri_{CS}$는 $H(H(r_{i',j} \| [m_{i',j}]_{O_i}) \| \cdots \| H(H(r_{i',j'}) \| [m_{i',j'}]_{O_i}))\ j \neq j'$에 의해 계산된 값이다. 여기에서 해시 함수 $H(\bullet)$의 입력은 s'개의 암호화된 블록들을 연결한 데이터이다. $[m_{i',j}]_{K_{O_i}}$는 트랩도어에 나열되어 있는 식별자들 중에서 $x(=(Veri_{CS} \bmod s'))+1$번째 식별자에 해당하는 암호화된 블록이다. TA는 $Veri_{TA} = H(V_{i',j} \| \cdots \| V_{i',j'})$을 계산한 다음에 $Veri_{TA} \fallingdotseq Veri_{CS}$가 성립하는지를 비교한다. 성립하지 않으면 소유자에게 통보하고, 성립한 경우에는 $(x+1)$번째 식별자 $H(m_{i',j})$와 CS로부터 수신한 $[m_{i',j}]_{K_{O_i}}$을 O_i에게 절차 9와 같이 전송한다.

[절차 9] $TA \rightarrow O_{i'}$: $[?, H(m_{i',j)}, [m_{i',j}]_{K_{O_i}}]_{K_{O_i,TA}}$,

$\forall i' \in U,\ 1 \le j \le s$

여기에서 '?'은 '감사'를 나타낸다.

O_i는 복호화된 블록 $m_{i',j}$을 해시 함수에 적용하여 $Veri_{O_i}(=H(m_{i',j}))$을 계산한 다음에 $Veri'_{O_i} \fallingdotseq H(m_{i',j})$가 성립하는지를 검사하여 그 결과를 TA에게 절차 10과 같이 전달한다.

[절차 10] $O_{i'} \rightarrow TA$: $[Veri_{O_i}]_{K_{O_i,TA}}$

V. 보안 및 성능 분석

1. 보안 분석

가. 정보 블록

정보소유자 O는 데이터 블록을 아웃소스하기 전에 자신만이 아는 비밀키로 암호화하기 때문에 키가 노출되지 않는 한 서버 CS에 저장된 중요한 정보(개인정보, 신체 정보)는 안전하게 유지된다. 비밀키가 노출되더라도 비밀키를 정기적 또는 비정기적(저장된 블록의 수정이나 사용자 Ur의 접근권한 취소 또는 소멸되는 경우)으로 다른 키로 변경하기 때문에 저장된 정보는 안전하다. 또한 다른 키로 변경하지 않더라도 혼합 해시 체인 방식에 의해서 키를 생성하기 때문에 노출된 이후에 생성한 블록들은 새로운 키를 받지 않는 한 안전하게 유지된다. 그리고 CS로 아웃소스한 블록의 식별자와 비밀키를 알고 있다고 하더라도 해당 블록의 내용이 수정되면 그 식별자도 변경되기 때문에 이전 식별자를 사용하여 더 이상 접근할 수 없다.

나. 신원 및 연관성

개인의 중요한 정보를 다루는 시스템에서 사용자의 신원이나 정보들 간의 연관성을 악의적인 공격자에게 노출되는 문제도 고려되어야 한다. 따라서 제안된 프로토콜에서는 정보 블록을 접근하는 O(또는 Ur)의 신원이 노출되는 문제와 블록을 검색이나 감사하는 과정에서 발생할 수 있는 문제들을 방지하는 구조를 제안하였다.

O가 자신의 소유 정보를 CS에게 아웃소스하거나 O(또는 Ur)가 CS로부터 정보를 다운로드할 때 직접 CS와 통신을 하지 않고 TA에게 위임하기 때문에 CS의 입장에서는 어떤 사용자가 아웃소스를 요청하는 것인지와 어떤 사용자가 정보에 접근하는지를 알 수 없기 때문에 O(또는 Ur)의 프라이버시가 보호될 수 있다. 그리고 O(또는 Ur)가 원하는 정보를 검색할 때 보안 강도를 나타내는 옵션(k)을 사용하고, TA는 옵션에 따른 트랩도어를 생성하여 CS에게 요청하기 때문에 CS는 요청하는 블록들 간의 연관성을 추론하는 것이 더 어렵게 된다. 또한 블록을 수정을 하는 경우에 블록의 다운로드와 아웃소싱, 삭제와 같은 절차를 거치는 동안 삭제 요청을 일정 시간 지연시킴으로써 삭제된 블록과 아웃소스한 노드와의 연관성 추론으로 야기될 수 있는 문제를 줄일 수 있다.

다. 무결성 감사

CS로 아웃소스한 정보의 무결성을 검사하기 위해 O나 TA가 모든 정보 블록들을 직접 다운로드하고 복호화한 후에 검사하는 것은 비효율적이며, 특히 스마트 패드나 스마트폰과 같은 모바일 기기를 사용하는 O들의 경우는 더 심각하다. 따라서 O

나 *TA*가 많은 블록들을 다운로드하고 복호화해야 하는 부담을 최소화할 수 있는 메커니즘이 요구된다.

본 논문에서는 효율적인 처리를 위해 *O*나 *TA*의 관여는 최소화하고 감사를 처리하는 대부분을 *CS*가 전담하는 해시 함수를 적용한 정보블록 식별자 기반 프로토콜을 제안하였으며, 이 제안 모델은 식별자로부터 원분을 생성할 수 없기 때문에 *TA*나 *CS*에게 정보가 노출되지 않는 장점을 갖는다.

2. 성능 분석

성능 분석을 위한 실제 시뮬레이션 환경은 3.0GHz에서 작동하는 듀얼 Intel Xeon CPU, 4GB 메모리 그리고 500명의 데이터 *O*에 대해 각 1,000개의 블록이 *CS*에 저장된 컴퓨터 환경에서 *O*와 *Ur*가 복록을 검색하는 과정을 통해 실험하였다. 여기에서 *TA*가 *CS*에게 요청하는 트랩도어내의 *O*의 블록의 수는 6~15개로 임의적으로 선택하였다.

그림 6은 *TA*가 존재하는 구조와 존재하지 않는 구조에서 300명의 *O*나 *Ur*가 *CS*에 저장된 블록을 다운로드하는데 소요되는 전체 시간을 나타낸 것이다. 전체 시간은 *TA*에서의 블록 인덱스 검색 시간과 트랩도어 생성시간, *CS* 블록 인덱스 검색 시간과 데이터 검색 시간, 전송시간을 포함한 것이다.

그림 6. 사용자 요청에 대한 전체 응답 시간

그림 6의 의하면 제안된 구조(Proposed)와 *TA*가 존재하지 않은 구조(without TA)에서 전체 소요되는 시간의 차이가 크지 않을 뿐만 아니라 대부분의 처리 시간이 *CS*에서 소요되기 때문에 실제 환경에 충분히 받아들일 수 있는 결과를 도출하였다.

VI. 결 론

정보 소유자(또는 정보 사용자)들과 클라우드 서버의 사이에 신뢰받는 에이전트 *TA*을 두어 정보 소유자(또는 정보사용자)들은 *CS*와 직접 통신하지 않고 *TA*와의 통신을 통해 메시지를 교환함으로써 정보 프라이버시의 침해를 최소화할 수 있는 구조를 제안하였다.

제안된 구조에서는 비밀키들을 효율적으로 관리하기 위하여 전방향 해시 함수와 후방향 해시 함수를 사용한 혼합 해시 체인 기법을 사용하였으며, 서버가 자신의 스토리지에 저장된 정보 검색을 요청하는 질의를 기반으로 해서 불법적으로 정보들 사이의 연관성 추론을 어렵게 하기 위해 검색 요청 질의에 보안 강도를 지정할 수 있는 옵션(k)을 사용하였다. 그리고 스토리지에 저장된 정보에 대한 소유자의 요청(수정, 삭제)에 대한 불이행이나 자체의 오작동으로 인한 과실(분실, 손상 등)의 문제를 해결하기 위해 감사 기능을 추가하였다. 그리고 보안과 성능 분석을 통해 제안된 u-Fitness 기반 운동관리 시스템이 실제 환경에서 적용 가능함을 보였다.

REFERENCES

[1] 신성훈, “웰니스 분야의 ICT 융합 기술 동향 및 전망,” *정보통신기술진흥센터, 기획시리즈*, 11-22쪽, 2016년

[2] S. K. Vashist, E. M. Schneider and John H.T. Luong, “Commercial Smartphone-Based Devices and Smart Applications for Personalized Healthcare Monitoring and Management,” *Diagnostics 2014* (http://www.mdpi.com/journal/diagnostics), pp. 104-128, 2014.

[3] D. D. Luxton, R, A. McCann, N. E. Bush, M. C. Mishkind and G. M. Reger, “mHealth for Mental Health: Integrating Smartphone Technology in Behavioral Healthcare,” *Professional Psychology: Research and Practice*, vol. 42, no. 6, pp. 505-512, 2011.

[4] 박성빈, 최준호, 김사엽, 형준호, 정경렬, “체력요인 DB를 활용한 u-Fitness 기반 맞춤형 운동 관리 시스템 개발,” *정보처리학회지*, 제19권 제4호, 56-63쪽, 2012년

[5] Sunyoung Kang, Seungae Kang, “Mobile exercise monitoring for personalized exercise prescription,” *Journal of The Korea Convergence Security Association*, vol. 15, no. 5, pp. 23-28, 2015.

[6] 지식경제 R&D 전략기획단, “2012 웰니스산업 동향 분석 및 발전방향,” *KITECH*, 2013년

[7] Dr. P. S. J. Kumar and Ms. A. S. Chaithra, “A Survey on Cloud Computing based Health Care for Diabetes: Analysis and Diagnosis,” *IOSR Journal of Computer Engineering*, vol. 17, Is. 4, Ver. I, pp. 109-117, 2015.

[8] L. Griebel, H. U. Prokosch, F. Köpcke, D. Toddenroth, J. Christoph, I. Leb, I. Engel and M. Sedlmayr, “A Scoping Review of Cloud Computing in Healthcare,” *BMC Medical Informatics and Decision Making*, https://

www.ncbi.nlm.nih.gov/pmc/articles/PMC4372226/, 2015.

[9] Yoon-Su Jeong, "Data Storage and Security Model for Mobile Healthcare Service based on IoT," *Journal of Digital Convergence*, vol. 15, no. 3, pp. 187-193, 2017.

[10] Li Chen, X. Sun, Z. Xia, and Q. Liu, "An Efficient and Privacy Semantic Multi-Keyword Ranked Search over Encrypted Cloud Data," *International Journal of Security and Its Application*, vol. 8, no. 2, pp. 323-332, 2014.

[11] D. Boneh, B. Lynn, and H. Shacham, "Short signatures from the Weil pairing," *in Advances in CryptologyASIACERYPT 2001*, ed: Springer, pp. 514-532, 2001.

[12] W. Zhang, Y. Lin, S. Xiao, J. Wu, and S. Zhou, "Privacy Preserving Ranked Multi-Keyword Search for Multiple Data Owners in Cloud Computing," *IEEE TRANSACTIONS ON COMPUTERS*, vol. 65, no. 5, pp. 1566-1577, 2016.

[13] DongGook Park, "Improving a Forward & Backward Secure Key Management Scheme for Wireless Sensor Networks," *International Journal of KIMICS*, vol. 7, no. 4, pp. 521-524, 2009.

저 자 소 개

김태연(정회원)

1988년 전남대학교 대학원 계산통계학과 석사 졸업.

1996년 전남대학교 대학원 전산통계학과 박사 졸업.

1996년~ 현재 서남대학교 컴퓨터정보학과 조교수.

<주관심분야 : 네트워크 보안, 이동컴퓨팅, 센서 네트워크, 미디어처리>

조기환(종신회원)

1987년 서울대학교 대학원 계산통계학과 석사 졸업.

1996년 영국 Newcastle대학교 전산학과 박사 졸업.

1987년~1997년 한국전자통신연구원 선임연구원.

1997년~1999년 목포대학교 컴퓨터과학과 전임강사.

1999년~ 현재 전북대학교 컴퓨터공학부 부교수.

<주관심분야 : 이동컴퓨팅, 센서네트워크, 상황인지 컴퓨팅, 분산처리 시스템>

최은복(정회원)

2000년 전남대학교 대학원 컴퓨터과학과 박사 졸업.

2000년~ 현재 전부대학교 스마트미디어학과 부교수.

<주관심분야 : 클라우드 컴퓨팅, 네트워크 관리, 미디어처리>