(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2019년03월26일
(11) 등록번호 10-1962347
(24) 등록일자 2019년03월20일

(51) 국제특허분류(Int. Cl.)
H04L 9/00 (2006.01) *H04L 29/06* (2006.01)
H04L 9/06 (2006.01) *H04L 9/08* (2006.01)
(52) CPC특허분류
H04L 9/008 (2013.01)
G06F 16/16 (2019.01)
(21) 출원번호 10-2017-0180765
(22) 출원일자 2017년12월27일
심사청구일자 2017년12월27일
(56) 선행기술조사문헌
Jian Liu et al., "Secure Deduplication of Encrypted Data without Additional Independent Servers", Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, ACM (2015)
US20170206218 A1

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
허준범
경기도 용인시 기흥구 보정로 30, 119동 201호(보정동, 동아솔레시티아파트)
신형준
제주특별자치도 서귀포시 대정읍 상모로290번길 18, 101동603호
(74) 대리인
김등용, 김홍석

전체 청구항 수 : 총 5 항 심사관 : 박보미

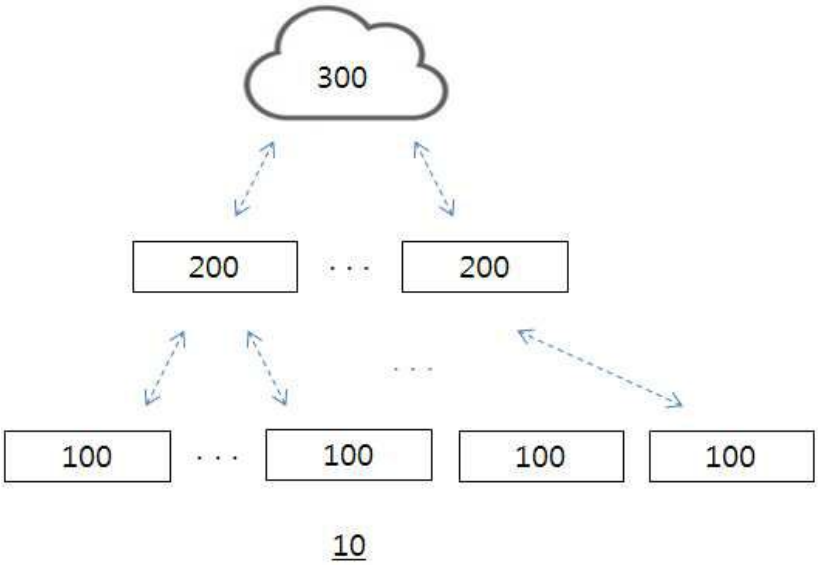
(54) 발명의 명칭 **포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법**

(57) 요 약

사용자 장치, 포그 디바이스, 및 클라우드 서버를 포함하는 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법이 개시된다. 하이브리드 데이터 중복제거 방법은 암호학적 해쉬 값(h) 및 짧은 길이의 해쉬 값(sh)생성하여 포그 디바이스에 송신하고, 포그 디바이스가 sh 를 클라우드 서버에 전달하는 단계, 클라우드 서버가 sh 와

(뒷면에 계속)

대 표 도 - 도1



10

같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치들의 집합을 검색하여 포그 디바이스에 송신하는 단계, 포그 디바이스가 상기 사용자 장치들의 집합 중에서 온라인 상태인 제2 사용자 장치를 탐색하는 단계, $siPAKE$ 프로토콜 실행하여 세션 키를 생성하는 단계, 포그 디바이스가 세션 키를 이용하여 데이터 중복을 탐지하고 암호화 키를 생성할 수 있는 값(e)을 생성하는 단계, 제1 사용자 장치가 제1 암호화 키 및 제1 암호문을 생성하는 단계, 포그 디바이스가 제1 태그 값을 생성하여 클라우드 서버에 전송하는 단계, 클라우드 서버가 제1 태그 값이 제2 태그 값과 동일한 지 확인하는 단계, 태그 값이 다른 경우, 클라우드 서버가 포그 디바이스에게 제1 암호문의 업로드를 요청하고, 제1 태그 값과 함께 저장하는 단계, 및 태그 값이 같은 경우, 이전에 제2 사용자 장치가 올린 제2 암호문에 대한 상기 제1 사용자 장치의 접근을 허용하는 단계를 포함한다.

(52) CPC특허분류
H04L 69/10 (2013.01)
H04L 9/0643 (2013.01)
H04L 9/0822 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 20150005650031001
부처명 과학기술정보통신부
연구관리전문기관 정보통신기술진흥센터
연구사업명 정보보호핵심원천기술개발사업
연구과제명 IoT 소프트웨어 보안 취약점 자동 분석 기술 개발
기 여 율 1/2
주관기관 고려대학교 산학협력단
연구기간 2017.06.01 ~ 2018.05.31
이 발명을 지원한 국가연구개발사업
과제고유번호 2017045199
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 (이공)중견연구자-핵심(후속)
연구과제명 분산 클라우드컴퓨팅을 위한 보안 및 프라이버시 계층 연구 및 개발
기 여 율 1/2
주관기관 고려대학교 산학협력단
연구기간 2017.06.01 ~ 2018.03.31

명 세 서

청구범위

청구항 1

복수의 사용자 장치, 복수의 포그 디바이스, 및 클라우드 서버를 포함하는 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 시스템에서의 안전하고 효율적인 하이브리드 데이터 중복제거 방법 방법에 있어서,

복수의 사용자 장치 중 데이터(F)를 클라우드 서버에 업로드 하고자 하는 제1 사용자 장치가 암호학적 해쉬 값($h = H(F)$) 및 짧은 길이의 해쉬 값($sh = SH(F)$)을 생성하고, 이를 상기 제1 사용자 장치와 연결된 포그 디바이스에 송신하는 단계;

상기 포그 디바이스가 전달받은 상기 짧은 길이의 해쉬 값(sh)을 클라우드 서버에 전달하는 단계;

상기 클라우드 서버가 상기 짧은 길이의 해쉬 값(sh)과 같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치들의 집합($\{U_i\}^{sh}$)을 검색하고, 검색된 사용자 장치들의 집합($\{U_i\}^{sh}$)을 상기 포그 디바이스에 송신하는 단계;

상기 포그 디바이스가 상기 사용자 장치들의 집합($\{U_i\}^{sh}$)에 속하는 사용자 장치 중에서 온라인 상태인 사용자 장치(이하, '제2 사용자 장치'라고 함)를 탐색하는 제2 사용자 장치 탐색 단계;

상기 제2 사용자 장치가 적어도 하나 이상 존재하는 경우, 상기 제1 사용자 장치는 $siPAKE$프로토콜 실행하여 상기 제2 사용자 장치 각각에 대응되는 제1 세션 키(k'_i)의 집합($\{k'_i\}$)을 생성하고, 상기 제2 사용자 장치 각각은 $siPAKE$ 프로토콜 실행하여 제2 세션 키(k_i)를 생성하는 $siPAKE$프로토콜 실행 단계;

상기 포그 디바이스가 상기 제1 세션 키(k'_i) 및 상기 제2 세션 키(k_i)를 이용하여 데이터 중복을 탐지하는 단계;

상기 포그 디바이스가 세션 키를 이용한 중복 탐지 결과에 따라 LWHE(Light-weight homomorphic encryption)를 통하여 암호화 키를 생성할 수 있는 값(e)을 생성하는 단계;

상기 제1 사용자 장치가 상기 포그 디바이스로부터 상기 암호화키를 생성할 수 있는 값(e)을 수신하고, 상기 암호화키를 생성할 수 있는 값(e)을 이용하여 제1 암호화 키($k_F = Dec(-r,e)$)를 생성하며, 상기 제1 암호화키(k_F)로 상기 데이터(F)를 암호화하여 제1 암호문($C = E(H_1(k_F),F)$)을 생성하는 단계;

상기 포그 디바이스가 상기 제1 암호문(C)에 대한 제1 태그 값($T = H(C)$)을 생성하고, 상기 제1 태그 값(T)을 상기 클라우드 서버에 전송하는 단계;

상기 클라우드 서버가 상기 제1 태그 값(T)이 이전에 제2 사용자 장치가 올린 제2 태그 값(T_j)과 동일한 지 확인하는 단계;

상기 제1 태그 값(T)과 제2 태그 값(T_j)이 다른 경우, 상기 클라우드 서버가 상기 포그 디바이스에게 상기 제1 암호문(C)의 업로드를 요청하고, 상기 제1 암호문(C)과 상기 제1 태그 값(T)을 저장하는 단계; 및

상기 제1 태그 값(T)과 제2 태그 값(T_j)이 같은 경우, 상기 클라우드 서버가 이전에 제2 사용자 장치가 올린 제2 암호문(C_j)에 대한 상기 제1 사용자 장치의 접근을 허용하고, 상기 제1 사용자 장치의 중복된 데이터 업로드를 방지하는 단계;를 포함하는,

포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법.

청구항 2

제1항에 있어서,

상기 제2 사용자 장치 탐색 단계 및 상기 $siPAKE$ 프로토콜 실행 단계는,

상기 사용자 장치들의 집합($\{U_i\}^{sh}$)에 상기 포그 디바이스와 연결된 제2 사용자 장치가 존재하는 경우, 상기 제1 사용자 장치 및 상기 제2 사용자 장치는 상기 포그 디바이스를 통해 $siPAKE$프로토콜을 실행하고,

상기 사용자 장치들의 집합($\{U_i\}^{sh}$)에 상기 포그 디바이스와 연결된 제2 사용자 장치가 존재하지 않는 경우, 상기 포그 디바이스는 다른 포그 디바이스들에게 다른 포그 디바이스들과 연결된 제2 사용자 장치의 탐색을 요청하며,

탐색 결과 다른 포그 디바이스들과 연결된 제2 사용자 장치가 존재할 경우, 상기 제1 사용자 장치 및 상기 제2 사용자 장치는 상기 제2 사용자 장치와 연결된 다른 포그 디바이스를 통해 $siPAKE$프로토콜을 실행하고,

탐색 결과 다른 포그 디바이스들과 연결된 제2 사용자 장치가 존재하지 않을 경우, 상기 포그 디바이스는 상기 제1 사용자 장치를 업로드 대상 데이터(F)에 대하여 최초의 사용자 장치로 인식하는,

포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법.

청구항 3

제1항에 있어서,

상기 $siPAKE$ 프로토콜 실행 단계에서,

상기 제1 사용자 장치는 상기 제1 사용자 장치의 암호학적 해쉬 값(h)을 패스워드로 사용하고,

상기 제2 사용자 장치는 상기 제2 사용자 장치의 암호학적 해쉬 값(h_i)을 패스워드로 사용하는,

포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법.

청구항 4

제1항에 있어서,

상기 제1 세션 키(k'_i) 및 상기 제2 세션 키(k_i)를 이용하여 데이터 중복을 탐지하는 단계는,

상기 제2 사용자 장치 각각이 의사난수 함수를 통하여 상기 제2 세션 키(k_i)의 길이를 확장하고, 확장된 제2

세션키를 $k_{iL}\|k_{iR}\ (k_{iR}\in Z)$ 로 나눈 후, k_{iL} 과 $Enc(k_{iR},k_{F_i})$ (이때, k_{F_i} 는 제2 사용자 장치가 이전에 데이터 F_i를 암호화할 때 사용한 암호화 키)를 생성하여 상기 포그 디바이스에 전달하는 과정;

상기 제1 사용자 장치가 상기 제1 세션키(k'_i) 각각에 대하여 의사난수 함수를 통하여 길이를 확장하고, 확장된 제1 세션키를 $k'_{iL}\|k'_{iR}$ 로 나눈 후, $\{k'_{iL}\}$ 과 $\{Enc(r,k'_{iR})\}$ (이때, r은 Z_p로부터 상기 제1 사용자 장치가 선택한 랜덤 원소)를 생성하여 상기 포그 디바이스에 전달하는 과정; 및

상기 포그 디바이스가 $k_{jL}=k'_{jL}$ 를 만족하는 인덱스 j가 존재하는지 탐색하는 과정을 포함하는,

포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법.

청구항 5

제4항에 있어서,

상기 암호화 키를 생성할 수 있는 값(e)을 생성하는 단계는,

$k_{jL}=k'_{jL}$ 를 만족하는 인덱스 j가 존재하는 경우, 상기 포그 디바이스가 아래와 같이 암호화 키를 생성할 수 있는 값(e)을 계산하는 과정; 및

$$e=Enc(k_{jR},k_{F_j})\ominus Enc(r,k'_{jR})=Enc(-r,k_{F_j})$$

(이때, $-r$ 은 순환 그룹 Z_p 에서 r 에 대한 덧셈에 대한 역수)

$k_{jL}=k'_{jL}$ 를 만족하는 인덱스 j가 존재하지 않을 경우, 상기 포그 디바이스가 순환 그룹 Z_p에서 임의의 r' 를 선택하고, 아래와 같이 암호화 키를 생성할 수 있는 값(e)을 생성하는 과정을 포함하는,

$$e=r'$$

포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법에 관한 것으로, 특히 데이터 업로더 사용자 장치가 암호 데이터를 포그 디바이스를 통하여 클라우드 스토리지에 업로드하는 환경에서 포그 디바이스에서 암호화된 데이터에 대한 비밀키 없이 중복을 탐지하고 제거하여 중복된 데이터의 업로드를 없애는 중복제거 방법에 관한 것이다.

배 경 기 술

[0002] 본 발명은 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 기법을 제안한다. 네트워크 끝 단인 클라이언트 근처에 포그 디바이스를 이용하여 높은 QoS(Quality of Service)와 낮은 지연시간을 제공할 수 있는 클라우드 컴퓨팅의 확장인 포그 컴퓨팅(Fog Computing) 환경은 현재 대량으로 발생하고 있는 데이터 저장소 및 통신 부하로 인해 현실적으로 효율적으로 사용되기 어려운 문제점이 있다. 또한, 서버 측에서 중복을 탐지하는 기존의 중복제거 기법들은 중복된 데이터의 업로드로 인한 비효율적인 추가비용이 발생하며, 클라이언트 측에서 미리 중

복을 탐지하여 중복된 데이터의 업로드를 제거한 기법들은 부 채널 공격에 취약하여 데이터 프라이버시가 보장되지 않는다.

[0003] 따라서, 따라서, 포그 컴퓨팅 환경에서 발생할 수 있는 상기와 같은 문제점을 해결하기 위해 포그 컴퓨팅 환경에 적합한 안전하고 효율적인 하이브리드 데이터 중복제거 방법이 필요하다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) US 9678973 B2

(특허문헌 0002) US 2017-0206218 A1

(특허문헌 0003) KR 2014-0141348 A

비특허문헌

[0005] (비특허문헌 0001) A Hybrid Deduplication for Secure and Efficient Data Outsourcing in Fog Computing (http://ieeexplore.ieee.org/abstract/document/7830695/)

발명의 내용

해결하려는 과제

[0006] 본 발명이 이루고자 하는 기술적인 과제는 서로 다른 사용자가 암호화된 데이터를 클라우드에 업로드하는 동시에 암호화된 데이터를 기반으로 중복을 제거를 함으로써 데이터 기밀성을 보장할 수 있고, 효율적인 클라우드 저장 공간의 관리가 가능한 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법을 제공하는 것이다.

[0007] 본 발명이 이루고자 하는 기술적인 과제는 포그 컴퓨팅 환경에 적합한 경량화된 동형암호(Light-weight homomorphic encryption ; LWHE)를 이용하여 낮은 지연시간 및 높은 QoS를 보장하는 하이브리드 데이터 중복제거 방법을 제공하는 것이다.

과제의 해결 수단

[0008] 본 발명의 실시 예에 따른 사용자 장치, 포그 디바이스, 및 클라우드 서버를 포함하는 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법은 하이브리드 데이터 중복제거 방법은 암호학적 해쉬 값(h) 및 짧은 길이의 해쉬 값(sh)생성하여 제1 사용자 장치와 연결된 포그 디바이스에 송신하고, 포그 디바이스가 이를 클라우드 서버에 전달하는 단계, 클라우드 서버가 상기 짧은 길이의 해쉬 값(sh)과 같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치들의 집합을 검색하여 포그 디바이스에 송신하는 단계, 포그 디바이스가 상기 사용자 장치들의 집합 중에서 온라인 상태인 제2 사용자 장치를 탐색하는 단계, $siPAKE$ 프로토콜 실행하여 세션 키를 생성하는 단계, 포그 디바이스가 세션 키를 이용하여 데이터 중복을 탐지하고 암호화 키를 생성할 수 있는 값(e)을 생성하는 단계, 제1 사용자 장치가 제1 암호화 키 및 제1 암호문을 생성하는 단계, 포그 디바이스가 제1 태그 값을 생성하여 클라우드 서버에 전송하는 단계, 클라우드 서버가 제1 태그 값이 제2 태그 값과 동일한 지 확인하는 단계, 태그 값이 다른 경우, 클라우드 서버가 포그 디바이스에게 제1 암호문의 업로드를 요청하고, 제1 태그 값과 함께 저장하는 단계, 및 태그 값이 같은 경우, 이전에 제2 사용자 장치가 올린 제2 암호문에 대한 상기 제1 사용자 장치의 접근을 허용하는 단계를 포함한다.

발명의 효과

[0009] 본 발명의 실시 예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법에 의할 경우, 서로 다른 사용자가 암호화된 데이터를 클라우드에 업로드하는 동시에 암호화된 데이터를 기반으로 중복제거를 함으로써 데이터 기밀성을 보장할 수 있고, 효율적인 클라우드 저장 공간의 관리가 가능하다.

[0010] 또한, 본 발명의 실시 예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법에 의할 경우, 포그 환경에 적합한 경량화된 동형암호 및 하이브리드 (클라이언트-포그디바이스 간 서버 측 중복제거, 포그디바이스-클라우드 간 클라이언트 측 중복제거) 중복제거 기법을 통하여 낮은 지연시간 및 높은 QoS를 보장할 수 있다.

도면의 간단한 설명

[0011] 본 발명의 상세한 설명에서 인용되는 도면을 보다 충분히 이해하기 위하여 각 도면의 상세한 설명이 제공된다.

도 1은 본 발명의 실시 예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 시스템의 개략도이다.

도 2는 도 1에 도시된 클라우드 서버에 암호문이 저장되는 구조를 도시한다.

도 3은 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법을 설명하기 위한 흐름도이다.

도 4 내지 도 6은 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법이 적용되는 시스템의 흐름을 도시한 도면이다.

발명을 실시하기 위한 구체적인 내용

[0012] 본 명세서에 개시되어 있는 본 발명의 개념에 따른 실시 예들에 대해서 특정한 구조적 또는 기능적 설명은 단지 본 발명의 개념에 따른 실시 예들을 설명하기 위한 목적으로 예시된 것으로서, 본 발명의 개념에 따른 실시 예들은 다양한 형태들로 실시될 수 있으며 본 명세서에 설명된 실시 예들에 한정되지 않는다.

[0013] 본 발명의 개념에 따른 실시 예들은 다양한 변경들을 가할 수 있고 여러 가지 형태들을 가질 수 있으므로 실시 예들을 도면에 예시하고 본 명세서에서 상세하게 설명하고자 한다. 그러나, 이는 본 발명의 개념에 따른 실시 예들을 특정한 개시 형태들에 대해 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물, 또는 대체물을 포함한다.

[0014] 제1 또는 제2 등의 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 상기 구성 요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성 요소를 다른 구성 요소로부터 구별하는 목적으로만, 예컨대 본 발명의 개념에 따른 권리 범위로부터 벗어나지 않은 채, 제1 구성 요소는 제2 구성 요소로 명명될 수 있고 유사하게 제2 구성 요소는 제1 구성 요소로도 명명될 수 있다.

[0015] 본 명세서에서 사용한 용어는 단지 특정한 실시 예를 설명하기 위해 사용된 것으로서, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 명세서에서, "포함하다" 또는 "가지다" 등의 용어는 본 명세서에 기재된 특징, 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성 요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.

[0016] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가진다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 갖는 것으로 해석되어야 하며, 본 명세서에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.

[0017] 이하, 본 명세서에 첨부된 도면들을 참조하여 본 발명의 실시 예들을 상세히 설명한다.

[0018] 우선, 본 발명에 대한 설명에 앞서, 본 발명의 배경이 되는 기술에 대해 설명하면 다음과 같다.

[0019] 패스워드 인증 기반 키 교환 프로토콜(Password Authenticated Key Exchange, PAKE): 동일한 Password를 가진 사용자사이 키 교환을 가능하게 한다. 본 발명에서는 Password를 업로드되는 데이터로부터 생성된 해쉬 값으로 사용함으로써, 같은 데이터 소유주 사이에 같은 세션 키를 공유가능하게 한다. 본 발명에서는 PAKE 프로토콜로 SPAKE2를 사용하며, 하나의 클라이언트가 여러 개의 클라이언트와 SPAKE2를 이용할 시 각 각의 PAKE 객체에 대해서 같은 입력 값을 사용한다는 기능이 보장된 $siPAKE$ 프로토콜(same-input-PAKE 프로토콜)을 활용한다.

[0020] 짧은 길이의 해쉬 함수(Short Hash Function, SH): 많은 충돌이 발생 가능한 짧은 길이의 해쉬를 생성하는 함수이다. 짧은 길이의 해쉬 함수로부터 생성된 값을 통하여 중복 확인의 효율성이 증가한다.

[0021] 동형암호(Homomorphic Encryption, HE): 비밀키 없이 암호화된 데이터들 사이의 연산을 통하여 기반 원본 데이터의 연산된 결과 값의 암호화된 결과물을 만드는 것을 가능하게 한다.

[0022] 경량화된 동형 암호화(Light-Weight Additively Homomorphic Encryption): 다음의 3가지 알고리즘으로 구성된다:

[0023] 1. $k \leftarrow Gen(1^{\lambda}, p)$: 키 생성 알고리즘으로, 암호화 키 k를 소수 p와 암호 파라미터 1^{λ}를 통해서 생성한다. k는 순환 그룹 Z_p의 랜덤 원소이다.

[0024] 2. $c \leftarrow Enc(k, m, p)$: 암호 알고리즘으로, 암호화 키 k, 소수 p, 평문 m을 입력으로 받아, 암호문 $c = m + k \pmod p$를 생성한다.

[0025] 3. $m \leftarrow Dec(k, c, p)$: 복호 알고리즘으로, 암호화 키 k, 소수 p, 암호문 c를 입력으로 받아, 평문 $m = c - k \pmod p$을 생성한다.

[0026] 이후 설명에서 입력으로 들어가는 소수 p는 생략한다.

[0027] 상기 경량화된 동형 암호화는 덧셈연산에 대한 동형적 성질을 가지고 있다. 구체적으로, $c_1 = Enc(k_1, m_1)$와 $c_2 = Enc(k_2, m_2)$에 대해서 $k = k_1 + k_2$ 일 때, $c_1 + c_2 = Enc(k, m_1 + m_2)$를 만족한다.

[0028] 이하, 도 1 내지 도 2를 참조하여, 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 시스템에 대해 상술한다.

[0029] 도 1은 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 시스템(10)을 도시하고, 도 2는 도 1에 도시된 클라우드 서버에 암호문이 저장되는 구조를 도시한다.

[0030] 도 1을 참조하면, 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 시스템(10)은 사용자 장치(100), 포그 디바이스(200), 및 클라우드 스토리지(300)를 포함한다.

[0031] 포그 디바이스(200)는 네트워크 망을 통하여 사용자 장치(100) 또는 클라우드 서버(300)와 통신할 수 있다. 일 실시예에 따른 네트워크 망은, 개방형 인터넷, 폐쇄형 인트라넷을 포함한 유선 인터넷망, 이동 통신망과 연동된 무선 인터넷 통신망뿐만 아니라, 각종 데이터 통신이 가능한 컴퓨터 네트워크 등의 가능한 통신 수단을 포함하는 넓은 개념이다.

[0032] 사용자 장치(100)는 데이터를 업로드 하고자 하는 업로더 사용자 장치(U_d, 이하, '제1 사용자 장치') 및 업로드 대상 데이터(F)에 대한 짧은 길이 해쉬 값 $sh = SH(F)$와 같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치(이하 '제2 사용자 장치')를 포함한다.

[0033] 사용자 장치(100)는 데이터(F)을 암호화하여 암호문(C)를 아래와 같이 생성한다. 포그 디바이스(200)를 통하여

데이터 중복 여부를 확인한 후, 중복되는 데이터가 존재하지 않으면 암호문(C)을 포그 디바이스(200)를 통하여 클라우드 서버(300)에 전송하여 저장한다.

[0034] $C_i = E(H_1(k_{F_i}), F_i)$ (이때, $k_{F_i} \in Z_p$)

[0035] 2는 도 1에 도시된 클라우드 서버(300)의 기능 블럭도이다.

[0036] 클라우드 서버(300)는 데이터 F_i에 대한 암호문 C_i, 암호문 C_i에 대응하는 짧은 길이의 해쉬 값 $sh_i = SH(F_i)$, 그리고 암호문 C_i에 대응하는 태그 값 $T_i = H(C_i)$을 도 2와 같은 형태로 저장한다.

[0037] 이하, 도 3 내지 도 4를 참조하여, 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법에 대하여 자세히 살펴보도록 한다.

[0038] 도 3은 도 1에 도시한 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법을 설명하기 위한 흐름도이다.

[0039] 도 4 내지 도 6은 본 발명의 일 실시예에 따른 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법이 적용되는 시스템의 흐름을 도시한 도면이다.

[0040] 우선, 임의의 길이의 문자열을 길이 n의 문자열로 변환하는 암호학적 해쉬 함수를 H_1이라 하고, E를 대칭암호(Symmetric key encryption)라고 할 때, 클라우드 서버(300)는 이전에 업로드된 암호문 $C_i = E(H_1(k_{F_i}), F_i)$ (이때, $k_{F_i} \in Z_p$) 각 각을 상기 암호문 C_i에 대응되는 짧은 길이의 해쉬 값 $sh_i = SH(F_i)$와 태그 값 $T_i = H(C_i)$을 도 2와 같은 형태로 저장한다.

[0041] 데이터(F)를 클라우드 서버(300, S)에 업로드하고자 하는 제1 사용자 장치(100, U_d)는 암호학적 해쉬 함수 H를 통해 암호학적 해쉬 값 $h = H(F)$를 생성하고, 짧은 길이의 해쉬 값 $sh = SH(F)$를 생성한다(S100). 제1 사용자 장치(100, U_d)는 짧은 길이의 해쉬 값 sh를 제1 사용자 장치(100, U_d)와 연결된 (근접) 포그 디바이스(200, Fog_d)에 전달하고, 포그 디바이스(200, Fog_d)는 전달받은 짧은 길이의 해쉬 값 sh를 클라우드 서버(300, S)에 전달한다.

[0042] 다음, 클라우드 서버(300, S)는 짧은 길이의 해쉬 값 sh와 같은 짧은 길이의 해쉬 값을 전송했던 사용자들의 집합 $\{U_i\}^{sh}$을 검색하고, 검색된 사용자 리스트를 포그 디바이스(200, Fog_d)에게 전달한다(S200).

[0043] 포그 디바이스(200, Fog_d)는 sh와 같은 짧은 길이의 해쉬 값을 전송했던사용자(또는 사용자 장치)들의 집합 $\{U_i\}^{sh}$을 수신하여, 포그 디바이스(200, Fog_d)와 연결된 사용자 장치 $U_i \in \{U_i\}^{sh}$를 탐색한다(S300).

[0044] 만약, 클라우드 서버(300)로부터 수신한 사용자 집합 $\{U_i\}^{sh}$에 해당 포그 디바이스(200, Fog_d)와 연결된 사용자 장치(제2 사용자 장치, U_i)가 존재할 경우, 업로더 사용자 장치(제1 사용자 장치, U_d)는 $siPAKE$

프로토콜을 포그 디바이스(200, Fog_d)를 통하여 진행한다.

[0045] 만약, 클라우드 서버(300)로부터 수신한 사용자 집합 $\{U_i\}^{sh}$에 해당 포그 디바이스(200, Fog_d)와 연결된 사용자 장치(제2 사용자 장치)가 존재하지 않을 경우, 해당 포그 디바이스(200, Fog_d)는 다른 포그 디바이스들에게 온라인 상태인 제2 사용자 장치(U_i)의 존재 여부에 대한 확인을 요청한다.

[0046] 만약, 다른 포그 디바이스들에게 온라인 상태인 제2 사용자 장치(U_i)가 존재할 경우, 제1 사용자 장치(U_d)는 온라인 상태인 제2 사용자 장치(U_i)가 존재하는 다른 포그 디바이스(200)를 통하여 $siPAKE$프로토콜을 진행한다.

[0047] 만약, 다른 포그 디바이스들에게 온라인 상태인 제2 사용자 장치(U_i)가 없을 경우, 즉, 포그 디바이스들과 연결되고 sh와 같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치(U_i)-가 존재하지 않을 경우, 포그 디바이스(Fog_d)는 상기 업로더 사용자 장치(제1 사용자 장치, U_d)를 데이터 F를 업로드한 최초의 사용자 장치로 인식한다.

[0048] 즉, 제1 사용자 장치(U_d)는 업로드 대상 데이터(F)에 대한 짧은 길이 해쉬 값 sh과 같은 짧은 길이의 해쉬 값을 전송했던 사용자 장치(U_i, 제2 사용자 장치)가 적어도 하나 이상 존재하는 경우 제2 사용자 장치와 연결된 포그 디바이스(Fog_d)를 통하여 $siPAKE$ 프로토콜을 수행한다.

[0049] $siPAKE$ 프로토콜을 수행할 때, 제1 사용자 장치(U_d)와 제2 사용자 장치(U_i)는 각각 제1 사용자 장치의 암호학적 해쉬 값 h와 제2 사용자 장치의 암호학적 해쉬 값 h_i를 패스워드로 사용한다.

[0050] 다음, 각 각의 제2 사용자 장치(U_i)는 $siPAKE$ 프로토콜 실행하여 제2 세션 키 k_i를 생성(S400)하고, 제1 사용자 장치(U_d)는 $siPAKE$ 프로토콜 실행하여 각 각의 다른 사용자 장치(즉, 제2 사용자 장치, U_i)에 대응되는 제1 세션 키의 집합 $\{k'_i\}$를 생성(S500)한다. 온라인 상태의 제2 사용자 장치가 m 개인 경우, m 개의 사용자 장치와 $siPAKE$ 프로토콜 수행하고, 그 결과물로 m 개의 제1 세션키 k'_i를 획득한다.

[0051] 만약, 제2 사용자 장치(U_i)가 업로드 한 데이터(F_i)와 제1 사용자 장치(U_d)가 업로드 할 데이터(F)가 동일($F_i = F$)하면, 제2 세션키와 대응하는 제1 세션키는 일치($k_i = k'_i$)한다. 즉, 포그 디바이스는 제1 세션키 및 제2 세션키를 이용하여 데이터 중복 여부를 확인한다.

[0052] 각 각의 제2 사용자 장치(U_i)는 제2 세션 키 k_i를 의사난수 함수를 통하여 길이를 확장하고, 길이가 확장된 k_i를 $k_{iL}||k_{iR}\ (k_{iR} \in Z)$로 나누고 k_{iL}과 $Enc(k_{iR}, k_{F_i})$(k_{F_i}는 제2 사용자 장치 U_i가 이전에 F_i를 암호화할 때 사용한 키)를 생성(S400)하여 포그 디바이스(Fog_d)에 전달한다.

[0053] 제1 사용자 장치(U_d)는 각 각의 제1 세션키 k'_i에 대하여 의사난수 함수를 통하여 길이를 확장하고, 제2 사용자 장치가 수행한 것과 같이 길이가 확장된 제1 세션키 k'_i를 $k'_{iL}||k'_{iR}$로 나누고 $\{k'_{iL}\}$과

$\{Enc(r,k'_{iR})\}$를 생성(S500)하여 포그 디바이스(Fog_d)에 전달한다. 이때, r은 Z_p로부터 제1 사용자 장치(U_d)가 선택한 랜덤 원소이다.

[0054] 다음, 포그 디바이스(Fog_d)는 제2 사용자 장치들 $\{U_i\}$로부터 k_{iL}과 $Enc(k_{iR},k_{F_i})$을 수신하고 제1 사용자 장치 U_d로부터 $\{k'_{iL}\}$과 $\{Enc(r,k'_{iR})\}$를 수신한 후 $k_{jL}=k'_{jL}$를 만족하는 인덱스 j가 존재하는지 확인하고, 그 결과에 따라 LWHE를 통하여 e(암호화키를 생성할 수 있는 값)를 계산한다(S600).

[0055] 만약, $k_{jL}=k'_{jL}$를 만족하는 인덱스 j가 존재하는 경우, 포그 디바이스(Fog_d)는 아래와 같이 e를 계산하여 제1 사용자 장치 U_d에 전송한다.

[0056] $$e=Enc(k_{jR},k_{F_j})\ominus Enc(r,k'_{jR})=Enc(-r,k_{F_j})$$

[0057] 이때, $-r$은 순환 그룹 Z_p에서 r에 대한 덧셈에 대한 역수이다.

[0058] 만약, $k_{jL}=k'_{jL}$를 만족하는 인덱스 j가 존재하지 않을 경우, 포그 디바이스(Fog_d)는 순환 그룹 Z_p에서 임의의 r'를 선택하고, $e=r'$를 제1 사용자 장치 U_d에 전송한다.

[0059] 다음, 제1 사용자 장치 U_d는 포그 디바이스로부터 e를 수신하여 제1 암호화키 $k_F=Dec(-r,e)$를 계산하고, 상기 제1 암호화키 k_F로 데이터 F를 암호화하여 제1 암호문 $C=E(H_1(k_F),F)$을 생성한다(S700). 이때, 포그 디바이스가 세션키를 통해 중복을 탐지한 경우 상기 제1 암호화키는 이전 중복 데이터를 암호화할 때 사용한 키와 동일하며, 세션키를 통해 중복이 탐지되지 않은 경우 상기 제1 암호화키는 랜덤하게 생성된 암호화키이다. 제1 사용자 장치 U_d는 포그 디바이스(Fog_d)에게 상기 제1 암호문 C를 전송한다(사용자 장치-포그디바이스 간 서버 측 중복제거 기법).

[0060] 포그 디바이스(Fog_d)는 제1 사용자 장치 U_d로부터 제1 암호문 C을 수신하여 상기 암호문 C에 대한 제1 태그 값 $T=H(C)$를 생성하고, 상기 제1 태그 값 T를 클라우드 서버(S)에게 전송한다(S800).

[0061] 클라우드 서버(S)는 상기 제1 태그 값 T과 이전 사용자(장치)가 올린 제2 태그 값 T_j이 같은지 확인한다. 즉, 제1 태그 값 T가 이전에 이미 저장되어 있는 태그 값인지 여부를 확인한다. 만약 제1 태그 값 T과 제2 태그 값 T_j이 동일한 경우, 즉, 중복된 태그 값이 탐지된 경우, 제1 사용자 장치 U_d의 이전 사용자(장치)가 올린 암호문 C_j에 대한 접근을 허용한다. 만약 제1 태그 값 T와 제2 태그 값 T_j가 다른 경우, 즉, 데이터의 중복이 탐지되지 아니한 경우, 클라우드 서버는 포그 디바이스(Fog_d)에게 제1 암호문 C의 업로드를 요청하고, 제1 암호문 C과 제1 태그 값 T를 저장한다(S900, 포그 디바이스-클라우드 서버 간 사용자 장치 측 중복제거).

[0062] 본 발명은 도면에 도시된 실시 예를 참고로 설명되었으나 이는 예시적인 것에 불과하며, 본 기술 분야의 통상의 지식을 가진 자라면 이로부터 다양한 변형 및 균등한 타 실시 예가 가능하다는 점을 이해할 것이다. 따라서, 본 발명의 진정한 기술적 보호 범위는 첨부된 등록청구범위의 기술적 사상에 의해 정해져야 할 것이다.

부호의 설명

[0063] 10 : 포그 컴퓨팅 환경에서의 하이브리드 데이터 중복제거 방법

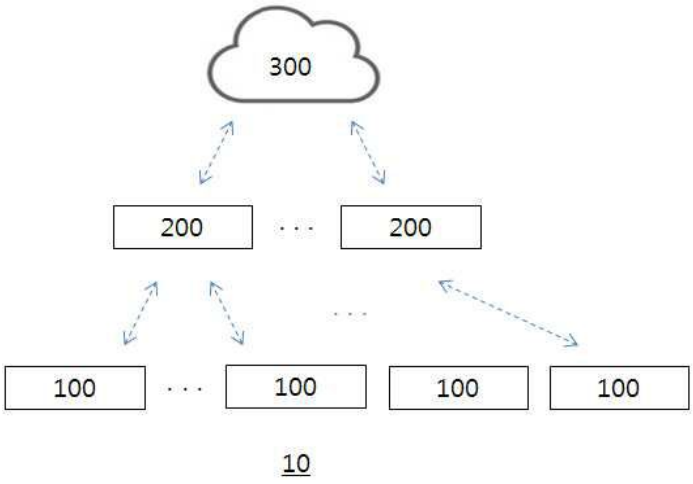
100 : 사용자 장치

200 : 포그 디바이스

300 : 클라우드 서버

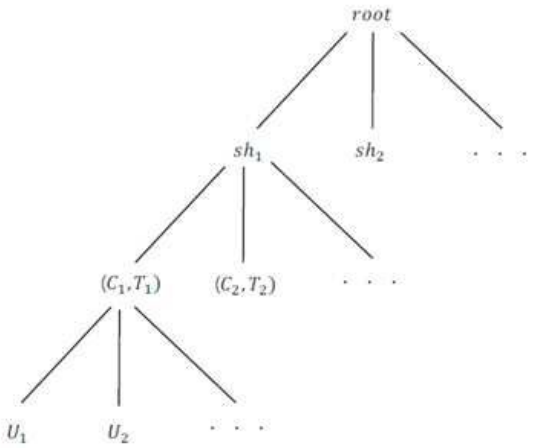
도면

도면1

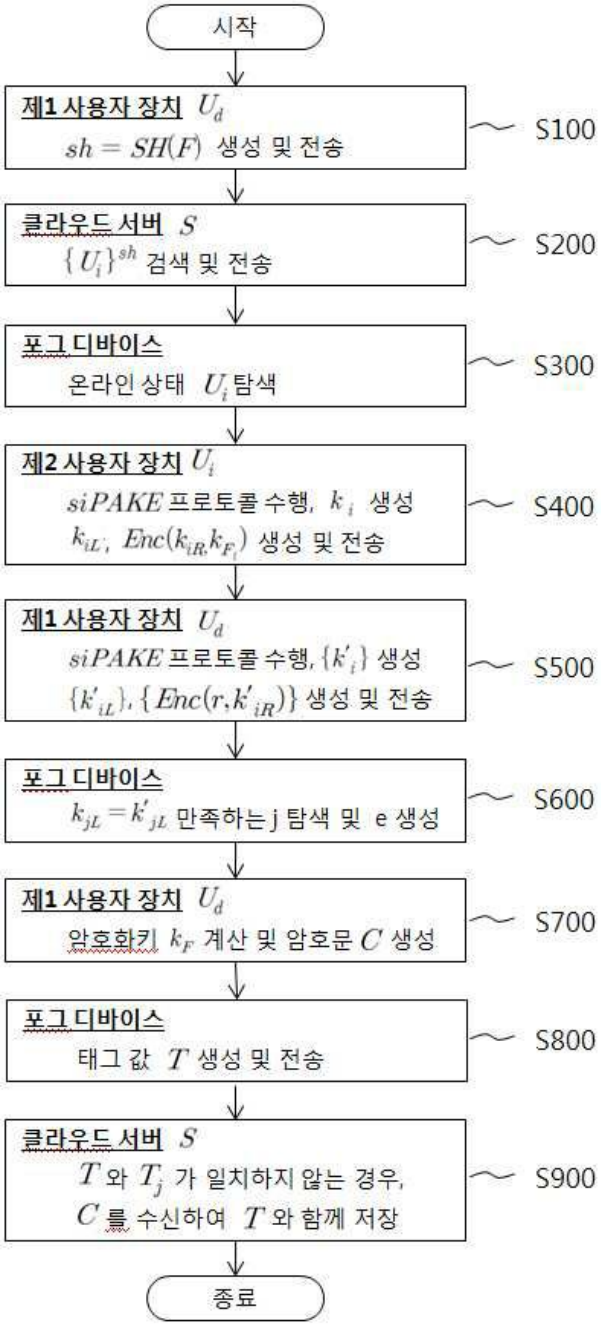


10

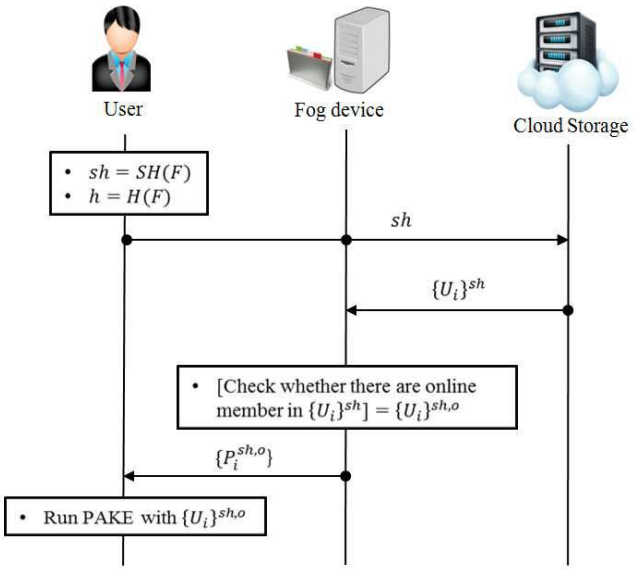
도면2



도면3

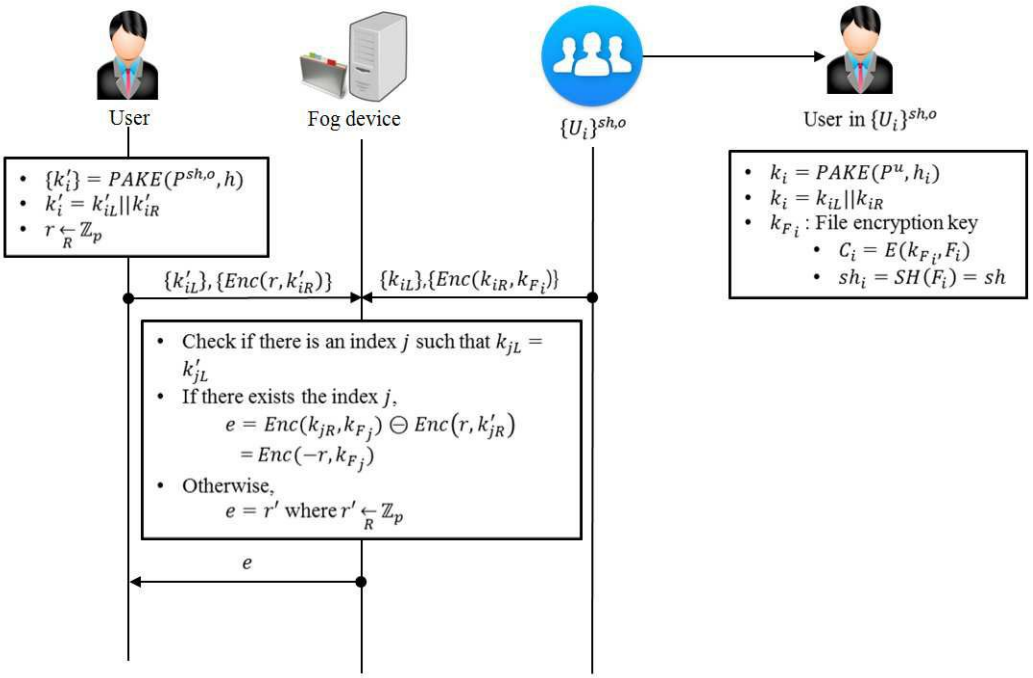

시작
제1 사용자 장치 U_d
$sh = SH(F)$ 생성 및 전송
S100
클라우드 서버 S
$\{U_i\}^{sh}$ 검색 및 전송
S200
포그 디바이스
온라인 상태 U_i 탐색
S300
제2 사용자 장치 U_i
$siPAKE$ 프로토콜 수행, k_i 생성
k_{iL}, $Enc(k_{iR}, k_{F_i})$ 생성 및 전송
S400
제1 사용자 장치 U_d
$siPAKE$ 프로토콜 수행, $\{k'_i\}$ 생성
$\{k'_{iL}\}$, $\{Enc(r, k'_{iR})\}$ 생성 및 전송
S500
포그 디바이스
$k_{jL} = k'_{jL}$ 만족하는 j 탐색 및 e 생성
S600
제1 사용자 장치 U_d
암호화키 k_F 계산 및 암호문 C 생성
S700
포그 디바이스
태그 값 T 생성 및 전송
S800
클라우드 서버 S
T 와 T_j 가 일치하지 않는 경우,
C 를 수신하여 T 와 함께 저장
S900
종료

도면4

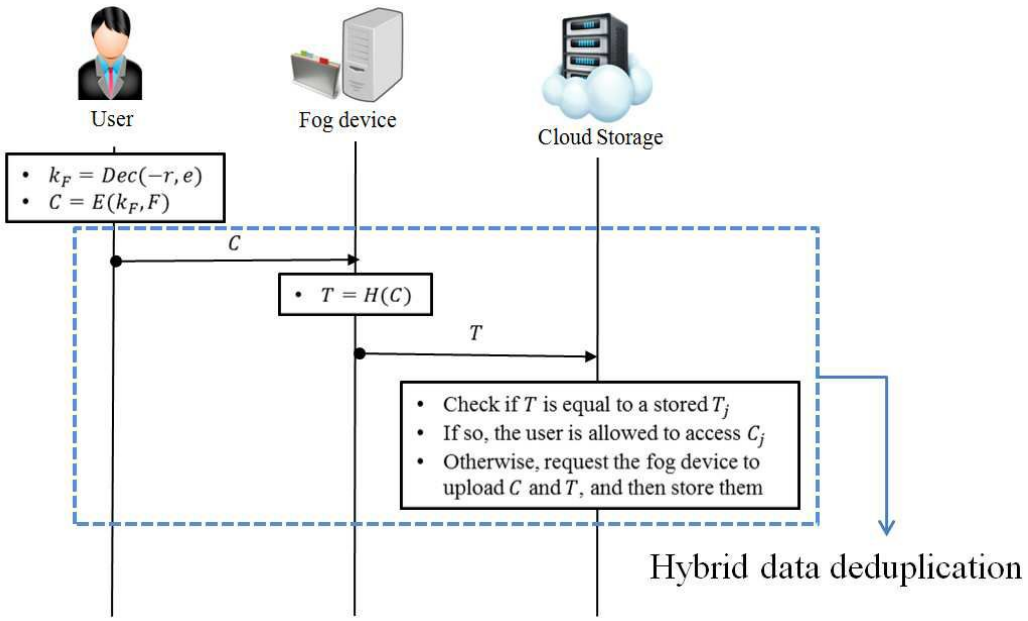


$*P_i^{sh,o}$: public parameter to run PAKE instance with online member i

도면5



도면6


User
Fog device
Cloud Storage
• $k_F = Dec(-r, e)$
• $C = E(k_F, F)$
C
• $T = H(C)$
T
• Check if T is equal to a stored T_j
• If so, the user is allowed to access C_j
• Otherwise, request the fog device to upload C and T, and then store them
Hybrid data deduplication