(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년02월15일
(11) 등록번호 10-1593674
(24) 등록일자 2016년02월03일

(51) 국제특허분류(Int. Cl.)
H04L 9/30 (2006.01)
(21) 출원번호 10-2014-0114171
(22) 출원일자 2014년08월29일
심사청구일자 2014년08월29일
(56) 선행기술조사문헌
KR101287597 B1
Dominique Schroder 외 1인, ACM Conference on Computer and Communications Security (CCS) 2012, "Verifiable Data Streaming", (2012.10.16. 공개)

(73) 특허권자
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
정익래
서울특별시 광진구 긴고랑로4길 53, 203호(중곡1동, 호동아파트)
김기성
서울특별시 성북구 안암로 145, 로봇융합관 404호 (안암동5가)
(74) 대리인
심경식, 홍성욱

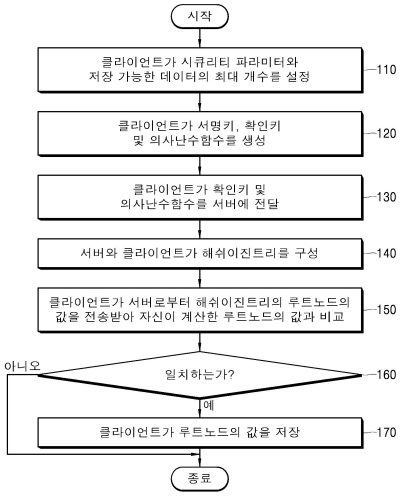
전체 청구항 수 : 총 15 항 심사관 : 문형섭

(54) 발명의 명칭 **검증 가능한 데이터 관리 방법 및 시스템**

(57) 요 약

본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 클라이언트가 시큐리티 파라미터와 저장 가능한 데이터의 최대 개수를 설정하는 단계; 상기 클라이언트가 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성하는 단계; 상기 클라이언트가 상기 확인키 및 상기 의사난수함수를 서버에 전달하는 단계; 상기 서버와 상기 클라이언트가 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성하는 단계; 상기 클라이언트가 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하는 단계; 및 상기 비교 결과 일치하면, 상기 클라이언트가 상기 루트노드의 값을 저장하는 단계를 포함한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 R1309081
부처명 미래창조과학부
연구관리전문기관 한국연구재단
연구사업명 (이공)중견연구자지원_핵심연구
연구과제명 [1차년도]실용적인 래티스 기반 익명성 보장 및 연산 가능한 서명 기술 연구
기 여 율 1/1
주관기관 고려대학교 산학협력단
연구기간 2013.12.01 ~ 2014.11.30

특허청구의 범위

청구항 1

클라이언트가 시큐리티 파라미터와 저장 가능한 데이터의 최대 개수를 설정하는 단계;

상기 클라이언트가 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성하는 단계;

상기 클라이언트가 상기 확인키 및 상기 의사난수함수를 서버에 전달하는 단계;

상기 서버와 상기 클라이언트가 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성하는 단계;

상기 클라이언트가 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하는 단계; 및

상기 비교 결과 일치하면, 상기 클라이언트가 상기 루트노드의 값을 저장하는 단계

를 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 2

제1항에 있어서,

상기 해쉬이진트리를 구성하는 단계는

임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출하는 단계; 및

상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성하는 단계

를 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 3

제1항에 있어서,

상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우,

상기 클라이언트가 상기 의사난수함수를 이용하여 난수 값을 계산하는 단계;

상기 클라이언트가 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 서버에 전송하는 단계;

상기 서버가 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증하는 단계; 및

상기 검증 결과 상기 서명이 정당하면, 상기 서버가 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장하는 단계

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 4

제1항에 있어서,

상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우,

상기 서버가 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 전송받는 단계;

상기 서버가 상기 인덱스를 이용하여 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색하는 단계; 및

상기 서버가 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색하여 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송하는 단계

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 5

제1항에 있어서,

상기 클라이언트가 i번째 데이터의 정당성을 검증하고자 하는 경우,

상기 클라이언트가 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증하는 단계;

상기 제1 검증 결과 정당한 서명인 경우, 상기 클라이언트가 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증하는 단계; 및

상기 제2 검증 결과 상기 난수의 값이 정당한 경우, 상기 클라이언트가 상기 i번째 데이터의 정당성을 인정하는 단계

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 6

제5항에 있어서,

상기 제2 검증하는 단계는

상기 클라이언트가 상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교하여 동일한지 여부를 확인하는 단계;

상기 클라이언트가 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교하여 동일한지 여부를 확인하는 단계; 및

상기 클라이언트가 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증하는 단계

를 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 7

제5항에 있어서,

상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우,

상기 i번째 데이터의 검증 결과 정당성이 인정되면, 상기 클라이언트가 상기 의사난수함수를 이용하여 새로운 난수 값을 계산하는 단계;

상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성하는 단계;

상기 클라이언트가 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경하는 단계;

상기 클라이언트가 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하는 단계;

상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 상기 서명, 및 상기 계산

된 루트노드의 값을 상기 서버에 전송하는 단계;

상기 서버가 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하는 단계; 및

상기 서버가 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터 전송받은 루트노드의 값을 비교하여 두 값이 동일하면 상기 i번째 데이터에 대한 업데이트를 완료하는 단계

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 8

제7항에 있어서,

상기 i번째 데이터의 업데이트 시,

상기 서버가 상기 난수 값의 변경이 발생한 노드로부터 루트노드에 이르는 상위노드들의 해쉬 값을 상기 변경된 새로운 난수 값에 따라 함께 업데이트하는 것을 특징으로 하는 검증 가능한 데이터 관리 방법.

청구항 9

클라이언트에 배치되고, 시큐리티 파라미터와 저장 가능한 데이터의 최대 개수를 설정하고, 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성하며, 상기 확인키 및 상기 의사난수함수를 서버에 전달하는 제1 데이터 설정부;

상기 서버 및 상기 클라이언트에 배치되고, 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성하는 해쉬이진트리 구성부; 및

상기 클라이언트에 배치되고, 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하고, 상기 비교 결과 일치하면, 상기 루트노드의 값을 저장하는 제2 데이터 설정부

를 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 10

제9항에 있어서,

상기 해쉬이진트리 구성부는

임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출하고, 상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 11

제9항에 있어서,

상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우,

상기 클라이언트에 배치되고, 상기 의사난수함수를 이용하여 난수 값을 계산하고, 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 서버에 전송하는 제1 데이터 업로드부; 및

상기 서버에 배치되고, 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증하고, 상기 검증 결과 상기 서명이 정당하면, 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째

노드에 저장되어 있는 노드 값과 함께 저장하는 제2 데이터 업로드부

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 12

제9항에 있어서,

상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우,

상기 서버에 배치되고, 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 전송받고, 상기 인덱스를 이용하여 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색하며, 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색하여 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송하는 데이터 검색부

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 13

제9항에 있어서,

상기 클라이언트가 i번째 데이터의 정당성을 검증하고자 하는 경우,

상기 클라이언트에 배치되고, 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증하고, 상기 제1 검증 결과 정당한 서명인 경우, 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증하며, 상기 제2 검증 결과 상기 난수의 값이 정당한 경우, 상기 i번째 데이터의 정당성을 인정하는 데이터 검증부

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 14

제13항에 있어서,

상기 데이터 검증부는

상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교하여 동일한지 여부를 확인하고, 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교하여 동일한지 여부를 확인하며, 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

청구항 15

제13항에 있어서,

상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우,

상기 클라이언트에 배치되고, 상기 i번째 데이터의 검증 결과 정당성이 인정되면, 상기 의사난수함수를 이용하여 새로운 난수 값을 계산하고, 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성하며, 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경하고, 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하며, 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 상기 서명, 및 상기 계산된 루트노드의 값을 상기 서버에 전송하는 제1 데이터 업데이트부; 및

상기 서버에 배치되고, 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하고, 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터

전송받은 루트노드의 값을 비교하여 두 값이 동일하면 상기 i번째 데이터에 대한 업데이트를 완료하는 제2 데이터 업데이트부

를 더 포함하는 것을 특징으로 하는 검증 가능한 데이터 관리 시스템.

명 세 서

기 술 분 야

[0001] 본 발명의 실시예들은 검증 가능한 데이터 기술에 관한 것으로서, 더욱 상세하게는 효율적인 검증 가능한 데이터 기술을 제공하는 데이터 관리 방법 및 시스템에 관한 것이다.

배 경 기 술

[0002] 검증 가능한 스트리밍 데이터(Verifiable Streaming Data) 기술은 클라이언트의 스트리밍 데이터에 대한 순서 유지, 무결성 보장 그리고 효율적인 저장/검색/업데이트를 제공하는 기술을 의미한다.

[0003] 신뢰할 수 없는 외부 스토리지에 자신의 민감한 데이터를 저장하는 환경은 이미 보편화되어 있다. 많은 사용자들이 다양한 웹하드 서비스에 자신의 민감한 데이터(사진, 동영상, 문서 등)를 업로드하여 사용하고 있으며, 병원 및 기업 등에서도 그 편의성 때문에 외부 스토리지 서비스를 사용하고 있다.

[0004] 이렇게 민감한 데이터 중에서 특히 스트리밍 데이터는 실시간으로 발생되고 저장되고 업데이트 되는 데이터로서, 최근 이슈가 되고 있는 유비쿼터스 환경, 카헬스 케어 시스템의 개발과 맞물려 급증하고 있는 데이터 형태이다.

[0005] 이미 신뢰할 수 없는 외부 스토리지에 클라이언트의 민감한 데이터를 안전하게 저장하고 관리할 수 있는 기술들은 많이 개발되어 왔다(Verifiable Database, Proofs of Retrievability). 하지만, 이러한 스트리밍 데이터에 대한 기술 개발은 상대적으로 적은 편이다.

[0006] 2012년 국제 저명 학술대회인 CCS '12(Proceedings of the 2012 ACM conference on Computer and communications security)에 Dominique Schroeder와 Heike Schroeder는 처음으로 Verifiable data streaming의 개념을 제시하고 기술을 설계하였다. 하지만, 그들의 기술은 클라이언트가 저장하고 있어야 하는 정보가 O(log N)이었고, 검색 및 업데이트를 하는 경우에도 O(log N)만큼의 연산량 및 전송량이 필요했다. 여기서 N이란 저장할 수 있는 최대 데이터의 수를 의미한다.

[0007] 따라서, 본 발명의 일 실시예에서는 그들이 제시한 기술의 한계점을 극복하고 보다 효율적으로 검증가능한 스트리밍 데이터 기술을 설계하는 방안을 제시하며, 최종적으로 클라이언트가 저장해야 하는 정보를 O(1)로 낮추고, 검색 및 업데이트 시에 발생하는 오버헤드 역시 O(log T)으로 낮출 수 있는 기술을 제안하고자 한다. 여기서, T는 저장할 수 있는 전체 데이터 N 보다 작거나 같은 값을 갖는 실제 저장한 데이터의 수를 의미한다.

[0008] 관련 선행기술로는 등록특허공보 제10-1369250호(발명의 명칭: P2P기반 네트워크에서 데이터의 무결성 검증을 위한 서버, 클라이언트 및 방법, 등록일자: 2014년 2월 25일)가 있다.

발명의 내용

해결하려는 과제

[0009] 본 발명의 일 실시예는 효율적으로 검증 가능한 데이터(스트리밍 데이터) 기술을 설계함으로써 데이터의 저장, 검색, 업데이트 시 발생하는 계산의 복잡도를 낮출 수 있도록 하는 검증 가능한 데이터 관리 방법 및 시스템을 제공한다.

[0010] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0011] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 클라이언트가 시큐리티 파라미터와 저장 가능한 데이터의 최대 개수를 설정하는 단계; 상기 클라이언트가 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성하는 단계; 상기 클라이언트가 상기 확인키 및 상기 의사난수함수를 서버에 전달하는 단계; 상기 서버와 상기 클라이언트가 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성하는 단계; 상기 클라이언트가 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하는 단계; 및 상기 비교 결과 일치하면, 상기 클라이언트가 상기 루트노드의 값을 저장하는 단계를 포함한다.

[0012] 상기 해쉬이진트리를 구성하는 단계는 임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출하는 단계; 및 상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성하는 단계를 포함할 수 있다.

[0013] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우, 상기 클라이언트가 상기 의사난수함수를 이용하여 난수 값을 계산하는 단계; 상기 클라이언트가 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 서버에 전송하는 단계; 상기 서버가 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증하는 단계; 및 상기 검증 결과 상기 서명이 정당하면, 상기 서버가 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장하는 단계를 더 포함할 수 있다.

[0014] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우, 상기 서버가 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 전송받는 단계; 상기 서버가 상기 인덱스를 이용하여 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색하는 단계; 및 상기 서버가 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색하여 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송하는 단계를 더 포함할 수 있다.

[0015] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 상기 클라이언트가 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증하는 단계; 상기 제1 검증 결과 정당한 서명인 경우, 상기 클라이언트가 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증하는 단계; 및 상기 제2 검증 결과 상기 난수의 값이 정당한 경우, 상기 클라이언트가 상기 i번째 데이터의 정당성을 인정하는 단계를 더 포함할 수 있다.

[0016] 상기 제2 검증하는 단계는 상기 클라이언트가 상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교하여 동일한지 여부를 확인하는 단계; 상기 클라이언트가 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교하여 동일한지 여부를 확인하는 단계; 및 상기 클라이언트가 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증하는 단계를 포함할 수 있다.

[0017] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법은 상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우, 상기 i번째 데이터의 검증 결과 정당성이 인정되면, 상기 클라이언트가 상기 의사난수함수를 이용하여 새로운 난수 값을 계산하는 단계; 상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성하는 단계; 상기 클라이언트가 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경하는 단계; 상기 클라이언트가 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하는 단계; 상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 상기 서명, 및 상기 계산된 루트노드의 값을 상기 서버에 전송하는 단계; 상기 서버가 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하는 단계; 및 상기 서버가 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터 전송받은 루트노드의 값을 비교하여 두 값이 동일하면 상기 i번째 데이터에 대한 업데이트를 완료하는 단계를 더 포함할 수 있다.

[0018] 상기 i번째 데이터의 업데이트 시, 상기 서버가 상기 난수 값의 변경이 발생한 노드로부터 루트노드에 이르는 상위노드들의 해쉬 값을 상기 변경된 새로운 난수 값에 따라 함께 업데이트할 수 있다.

[0019] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템은 클라이언트에 배치되고, 시큐리티 파라미터와 저장 가능한 데이터의 최대 개수를 설정하고, 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성하며, 상기 확인키 및 상기 의사난수함수를 서버에 전달하는 제1 데이터 설정부; 상기 서버 및 상기 클라이언트에 배치되고, 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성하는 해쉬이진트리 구성부; 및 상기 클라이언트에 배치되고, 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하고, 상기 비교 결과 일치하면, 상기 루트노드의 값을 저장하는 제2 데이터 설정부를 포함한다.

[0020] 상기 해쉬이진트리 구성부는 임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출하고, 상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성할 수 있다.

[0021] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템은 상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우, 상기 클라이언트에 배치되고, 상기 의사난수함수를 이용하여 난수 값을 계산하고, 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 서버에 전송하는 제1 데이터 업로드부; 및 상기 서버에 배치되고, 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증하고, 상기 검증 결과 상기 서명이 정당하면, 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장하는 제2 데이터 업로드부를 더 포함할 수 있다.

[0022] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템은 상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우, 상기 서버에 배치되고, 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 전송받고, 상기 인덱스를 이용하여 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색하며, 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색하여 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송하는 데이터 검색부를 더 포함할 수 있다.

[0023] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템은 상기 클라이언트에 배치되고, 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증하고, 상기 제1 검증 결과 정당한 서명인 경우, 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증하며, 상기 제2 검증 결과 상기 난수의 값이 정당한 경우, 상기 i번째 데이터의 정당성을 인정하는 데이터 검증부를 더 포함할 수 있다.

[0024] 상기 데이터 검증부는 상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교하여 동일한지 여부를 확인하고, 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교하여 동일한지 여부를 확인하며, 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증할 수 있다.

[0025] 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템은 상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우, 상기 클라이언트에 배치되고, 상기 i번째 데이터의 검증 결과 정당성이 인정되면, 상기 의사난수함수를 이용하여 새로운 난수 값을 계산하고, 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성하며, 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경하고, 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하며, 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 상기 서명, 및 상기 계산된 루트노드의 값을 상기 서버에 전송하는 제1 데이터 업데이트부; 및 상기 서버에 배치되고, 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산하고, 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터 전송받은 루트노드의 값을 비교하여 두 값이 동일하면 상기 i번째 데이터에 대한 업데이트를 완료하는 제2 데이터 업데이트부를 더 포함할 수 있다.

[0026] 기타 실시예들의 구체적인 사항들은 상세한 설명 및 첨부 도면들에 포함되어 있다.

발명의 효과

[0027] 본 발명의 일 실시예에 따르면, 클라이언트가 저장해야 하는 정보량을 O(1)로 낮춤으로써 클라이언트가 별도로 저장해야 하는 정보의 양이 외부 서버에 저장하는 데이터의 총 개수와 상관없도록 하며, 이를 통해 저장 공간 효율성을 향상시킬 수 있다.

[0028] 본 발명의 일 실시예에 따르면, 클라이언트가 데이터를 검색 및 업데이트 시 발생하는 연산 및 전송 오버헤드 역시 O(log T)으로 낮출 수 있는데, 이 역시 저장할 수 있는 총 데이터의 양에 따라 증가하는 것이 아니라 현재 저장된 데이터의 양에 따라 증가하는 것이기 때문에 기존의 기술들에 비해 매우 유리하다.

[0029] 본 발명의 일 실시예에 따르면, 데이터를 서버에 효율적으로 저장할 수 있도록 설계하여 실시간으로 발생하는 스트리밍 데이터에 매우 적합하다.

도면의 간단한 설명

[0030] 도 1은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터를 설정하는 방법을 설명하기 위해 도시한 흐름도이다.

도 2는 본 발명의 일 실시예에 따라 해쉬이진트리를 구성하는 방법을 설명하기 위해 도시한 흐름도이다.

도 3은 본 발명의 일 실시예에 따라 구성되는 해쉬이진트리를 도시한 도면이다.

도 4는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 업로드 방법을 설명하기 위해 도시한 흐름도이다.

도 5는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 검색 방법을 설명하기 위해 도시한 흐름도이다.

도 6은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 검증 방법을 설명하기 위해 도시한 흐름도이다.

도 7은 본 발명의 일 실시예에 따라 난수 값에 대한 정당성을 검증하는 방법을 설명하기 위해 도시한 흐름도이다.

도 8은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 업데이트 방법을 설명하기 위해 도시한 흐름도이다.

도 9 내지 도 14는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템을 설명하기 위해 도시한 블록도이다.

발명을 실시하기 위한 구체적인 내용

[0031] 본 발명의 이점 및/또는 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 상세하게 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나, 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 것이며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하며, 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다. 명세서 전체에 걸쳐 동일 참조 부호는 동일 구성요소를 지칭한다.

[0032] 검증 가능한 스트리밍 데이터 (Verifiable Streaming Data) 기술이란 신뢰할 수 없는 외부 저장 공간, 즉 외부 스토리지 서비스에 클라이언트의 민감한 스트리밍 데이터를 저장하는 경우 다음과 같은 보안 요구사항을 충족할 수 있는 기술을 의미한다.

[0033] 1. 스트리밍 데이터의 저장/검색/수정이 효율적이어야 한다.

[0034] 2. 스트리밍 데이터의 무결성 보장 즉, 클라이언트는 자신이 검색한 데이터가 실제로 자신이 저장한 데이터인지 확신할 수 있어야하며, 나아가 최신 업데이트 데이터인지 확인할 수 있어야 한다.

[0035] 3. 스트리밍 데이터에 대한 순서를 유지 및 확인할 수 있어야 한다.

[0036] 본 발명은 이러한 보안 요구사항을 만족하면서 동시에 가장 효율적인 검증 가능한 스트리밍 데이터 기술에 관한 것이다.

[0037] 이하에서는 첨부된 도면을 참조하여 본 발명의 실시예들을 상세히 설명하기로 한다.

[0038] 도 1은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법을 설명하기 위해 도시한 흐름도로서, 특히 데이터를 설정(Setup)하는 방법을 설명하기 위한 도면이다.

[0039] 도 1을 참조하면, 단계(110)에서 클라이언트가 시큐리티 파라미터(security parameter)와 저장 가능한 데이터의 최대 개수를 설정한다.

[0040] 다음으로, 단계(120)에서 상기 클라이언트가 상기 설정된 시큐리티 파라미터에 기초하여 서명키(비밀키), 확인키(공개키) 및 의사난수함수를 생성한다.

[0041] 다음으로, 단계(130)에서 상기 클라이언트가 상기 확인키 및 상기 의사난수함수를 서버에 전달한다.

[0042] 다음으로, 단계(140)에서 상기 서버와 상기 클라이언트가 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성한다.

[0043] 이하에서는 상기 해쉬이진트리를 구성하는 단계(140)에 대해 도 2 및 도 3을 참조하여 구체적으로 설명한다. 참고로, 도 2는 본 발명의 일 실시예에 따라 해쉬이진트리를 구성하는 방법을 설명하기 위해 도시한 흐름도이고, 도 3은 본 발명의 일 실시예에 따라 구성되는 해쉬이진트리를 도시한 도면이다.

[0044] 도 2 및 도 3을 참조하면, 단계(210)에서 상기 서버와 상기 클라이언트는 임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출할 수 있다.

[0045] 이어서, 단계(220)에서 상기 서버와 상기 클라이언트는 상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성할 수 있다.

[0046] 다시 도 1을 참조하면, 단계(150)에서 상기 클라이언트가 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교한다.

[0047] 상기 비교 결과 일치하면(160의 "예" 방향), 단계(170)에서 상기 클라이언트가 상기 루트노드의 값을 저장한다.

[0048] 상기 데이터를 설정(Setup)하는 방법에 대해 보다 구체적으로 설명하면 다음과 같다.

[0049] 상기 클라이언트는 시큐리티 파라미터 k와 저장할 수 있는 데이터의 최대 개수 2^D-1를 설정한다. 상기 클라이언트는 상기 설정된 시큐리티 파라미터 k에 따라 전자서명 알고리즘에 사용될 서명키 SK와 확인키 PK를 생성하고, 의사난수함수 $f:Z\rightarrow\{0,1\}^k$를 생성한다. 상기 클라이언트는 상기 생성된 확인키 PK와 의사난수함수 f를 서버에게 전달한다.

[0050] 상기 서버와 상기 클라이언트는 도 3과 같이 안전한 해쉬함수, 즉 출동저항성을 갖는 해쉬함수 H를 사용하여 해쉬이진트리를 구성한다. 도 3에서 각 난수는 $r_i=f(i)$으로 계산되고, $r'_i = r_i \| H(r'_{2i}) \| H(r'_{2i+1})$ 으로 정의되며, $i=2^{d-1},\ldots,2^d-1$, 즉 최하위 노드의 값에 대해서만 도 3에서와 같이 $r'_i = r_i$으로 연산한다.

[0051] 마지막으로, 상기 서버는 루트노드의 값인 $r_1, H(r'_2), H(r'_3)$를 상기 클라이언트에게 보내게 되고, 상기 클라이언트는 자신이 계산한 루트노드의 값과 비교하여 일치하면 이를 인증하고 저장하게 된다. 즉, 상기 서버는 도 3의 해쉬이진트리와 확인키 PK를 저장하게 되고, 상기 클라이언트는 $f, SK, r_1, H(r'_2), H(r'_3)$을 저장하게 된다.

[0052] 도 4는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 업로드 방법을 설명하기 위해 도시한 흐름도이다. 즉, 도 4는 상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우

에 대한 것이다.

[0053] 도 4를 참조하면, 단계(410)에서 상기 클라이언트가 상기 의사난수함수를 이용하여 난수 값을 계산한다.

[0054] 다음으로, 단계(420)에서 상기 클라이언트가 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 서버에 전송한다.

[0055] 다음으로, 단계(430)에서 상기 서버가 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증한다.

[0056] 다음으로, 단계(440)에서 상기 검증 결과 상기 서명이 정당하면(450의 "예" 방향), 단계(460)에서 상기 서버가 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장한다.

[0057] 상기 데이터를 업로드(Append)하는 방법에 대해 보다 구체적으로 설명하면 다음과 같다.

[0058] 상기 클라이언트가 자신의 i번째 데이터 m_i을 서버에 저장하고자 하는 경우, 우선 난수 값 $r_i=f(i)$을 계산하고, 데이터 $m_i' = r_i \| m_i$에 대한 서명 σ_i을 자신의 서명키 SK를 이용해 생성하여 상기 서버에게 전송한다.

[0059] 상기 서버는 해당 메시지에 대해 서명이 정당한지를 확인키 PK를 이용해 검증하고, (m_i', σ_i)를 자신의 해쉬이진트리 i번째 노드에 이미 저장되어 있는 $r_i, H(r'_{2i}), H(r'_{2i+1})$와 함께 저장한다.

[0060] 이와 같이, 본 발명의 일 실시예에서는 데이터를 저장함에 있어서, 단순히 클라이언트의 서명 생성만을 사용할 뿐, 어떠한 추가적인 연산도 필요로 하지 않는다. 이는 매우 효율적인 데이터 저장이 가능함을 의미하며, 특히 실시간으로 발생하는 스트리밍 데이터에 매우 적합한 방식임을 알 수 있다.

[0061] 도 5는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 검색 방법을 설명하기 위해 도시한 흐름도이다. 즉, 도 5는 상기 클라이언트가 자신의 i번째 데이터를 검색하고자 하는 경우에 대한 것이다.

[0062] 도 5를 참조하면, 단계(510)에서 상기 서버가 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 수신한다.

[0063] 다음으로, 단계(520)에서 상기 서버가 상기 인덱스를 이용하여 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색한다.

[0064] 다음으로, 단계(530)에서 상기 서버가 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색한다.

[0065] 다음으로, 단계(540)에서 상기 서버가 상기 탐색된 난수 값 및 해쉬 값을 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송한다.

[0066] 상기 데이터를 검색(Search)하는 방법에 대해 보다 구체적으로 설명하면 다음과 같다.

[0067] 상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우, 먼저 인덱스 i를 서버에게 전송하면, 상기 서버는 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 (m_i', σ_i)와 함께 루트노드로 가는 자신의 상위 노드들에 저장되어 있는 난수 값 및 해쉬 값들을 상기 클라이언트에게 검증값 π_i으로 전송한다.

[0068] 즉, $1 < \lfloor i/2^d \rfloor$을 만족하는 가장 큰 양의 정수를 c라고 했을 때, π_i는 d=0,1,...,c에 대해 $r_{\lfloor i/2^d \rfloor}, H(r'_{2\lfloor i/2^d \rfloor}), H(r'_{2\lfloor i/2^d \rfloor+1})$의 값들을 의미한다. 따라서, 상기 서버가 상기 클라이언트에게 보내주는 값들은, 전체 저장 가능한 데이터의 개수에 따라 증가하는 것이 아니라, 현재 저장되어 있는 데이터의 수에 따라 증가하게 된다.

[0069] 보다 자세하게는 도 3의 해쉬이진트리에서 각 층마다 1개의 값을 돌려주는 구조이기 때문에 O(log T)만큼의 데이터를 상기 클라이언트에게 전송하게 된다. 여기서, T는 현재 저장되어 있는 데이터의 수를 의미한다.

[0070] 도 6은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 검증 방법을 설명하기 위해 도시한 흐름도이다.

[0071] 도 6을 참조하면, 단계(610)에서 상기 클라이언트가 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증한다.

[0072] 상기 제1 검증 결과 정당한 서명인 경우(620의 "예" 방향), 단계(630)에서 상기 클라이언트가 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증한다.

[0073] 이하에서는 상기 제2 검증 과정에 대해 도 7을 참조하여 구체적으로 설명한다. 참고로, 도 7은 본 발명의 일 실시예에 따라 난수 값에 대한 정당성을 검증하는 방법을 설명하기 위해 도시한 흐름도이다.

[0074] 도 7을 참조하면, 단계(710)에서 상기 클라이언트가 상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교한다.

[0075] 상기 제1 비교 결과 두 값이 동일하면(720의 "예" 방향), 단계(720)에서 상기 클라이언트가 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교한다.

[0076] 상기 제2 비교 결과 두 값이 동일하면(740의 "예" 방향), 단계(750)에서 상기 클라이언트가 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증한다. 즉, 상기 클라이언트는 제1 및 제2 비교 결과 모두 동일하면, 상기 난수 값에 대해 정당한 것으로 판단할 수 있다.

[0077] 다시 도 6을 참조하면, 상기 제2 검증 결과 상기 난수의 값이 정당한 것으로 검증되면(640의 "예" 방향), 단계(650)에서 상기 클라이언트가 상기 i번째 데이터의 정당성을 인정한다.

[0078] 상기 데이터를 검증(Verify)하는 방법에 대해 보다 구체적으로 설명하면 다음과 같다.

[0079] 상기 클라이언트는 가장 먼저 상기 서버에게 받은 (m_i', σ_i)이 정당한 서명인지를 검증한다. 해당 검증 과정을 통과하게 되면, $m_i' = r_i \| m_i$에 포함된 r_i의 값에 대한 정당성을 다음과 같이 검증하게 된다.

[0080] 즉, 상기 클라이언트는 d=0,1,...,c에 대해 $H(r_{\lfloor i/2^d \rfloor} \| H(r'_{2\lfloor i/2^d \rfloor}) \| H(r'_{2\lfloor i/2^d \rfloor + 1}))$의 값이 $H(r'_{\lfloor i/2^d \rfloor})$값과 동일한지 여부를 확인한다. 마지막으로, 상기 클라이언트는 d=c인 경우에는 $H(r_{\lfloor i/2^d \rfloor} \| H(r'_{2\lfloor i/2^d \rfloor}) \| H(r'_{2\lfloor i/2^d \rfloor + 1}))$의 값이 자신이 소유하고 있는 $H(r'_{\lfloor i/2^d \rfloor})$의 값과 동일한지 여부를 확인함으로써 r_i에 대한 정당성을 확인한다.

[0081] 상기 클라이언트는 해당 서명 값이 정당하고 동시에 그 안에 포함된 난수 값이 정당할 경우에만 해당 데이터의 정당성을 인정한다.

[0082] 도 8은 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 방법에 있어서, 데이터 업데이트 방법을 설명하기 위해 도시한 흐름도이다. 즉, 도 8은 상기 클라이언트가 자신의 i번째 데이터를 업데이트하고자 하는 경우에 대한 것이다.

[0083] 도 8을 참조하면, 상기 i번째 데이터의 검증 결과 정당성이 인정되면(도 6 참조), 단계(810)에서 상기 클라이언트가 상기 의사난수함수를 이용하여 새로운 난수 값을 계산한다.

[0084] 다음으로, 단계(820)에서 상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성한다.

[0085] 다음으로, 단계(830)에서 상기 클라이언트가 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경한다.

[0086] 다음으로, 단계(840)에서 상기 클라이언트가 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산한다.

[0087] 다음으로, 단계(850)에서 상기 클라이언트가 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터,

상기 서명, 및 상기 계산된 루트노드의 값을 상기 서버에 전송한다.

[0088] 다음으로, 단계(860)에서 상기 서버가 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산한다.

[0089] 다음으로, 단계(870)에서 상기 서버가 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터 전송받은 루트노드의 값을 비교한다.

[0090] 상기 비교 결과 두 값이 동일하면(880의 "예" 방향), 단계(890)에서 상기 서버가 상기 i번째 데이터에 대한 업데이트를 완료한다.

[0091] 상기 데이터를 업데이트(Update)하는 방법에 대해 보다 구체적으로 설명하면 다음과 같다.

[0092] 상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우, 먼저 Search 알고리즘(도 5 참조)을 사용하여 i번째 데이터와 그에 대한 π_i값을 서버로부터 받아 이를 Verify 알고리즘(도 6, 7 참조)으로 검증한다.

[0093] 검증이 통과한 경우, 상기 클라이언트는 자신의 새로운 i번째 데이터 $\overline{m_i}$을 상기 서버에 저장하기 위해, 새로운 난수값 $\overline{r_i}$을 선택하고, 데이터 $\overline{m_i'} = \overline{r_i} \| \overline{m_i}$에 대한 서명 $\overline{\sigma_i}$을 생성한다.

[0094] 그 후, 상기 해쉬이진트리의 i번째 노드에 있는 r_i을 $\overline{r_i}$으로 변경하고, π_i에 포함된 각 노드들의 값을 상기 변경된 $\overline{r_i}$로부터 새로 연산하여 새로운 루트노드의 값을 계산한다.

[0095] 즉, d=1,...,c에 대해 $H(r'_{\lfloor i/2^d \rfloor})$의 값을 새로 계산된 $H(r_{\lfloor i/2^d \rfloor} \| H(r'_{2\lfloor i/2^d \rfloor}) \| H(r'_{2\lfloor i/2^d \rfloor+1}))$으로 변경하고, 마지막으로 d=c인 경우에는 클라이언트 자신이 소유하고 있는 $H(r'_{\lfloor i/2^c \rfloor})$의 값을 새로 연산된 $H(r_{\lfloor i/2^c \rfloor} \| H(r'_{2\lfloor i/2^c \rfloor}) \| H(r'_{2\lfloor i/2^c \rfloor+1}))$으로 업데이트한다.

[0096] 이를 상기 서버에 반영하기 위해서, 상기 클라이언트는 모든 중간 값을 전송할 필요 없이, 새로 생성한 $(\overline{m_i'} = \overline{r_i} \| \overline{m_i}, \overline{\sigma_i})$ 및 루트노드의 값만을 전송하고, 상기 서버는 상기 클라이언트와 같은 방법으로, 업데이트된 노드로부터 루트노드의 값을 새롭게 연산하여 상기 클라이언트로부터 받은 루트노드의 값과 비교하여 같은 경우에 업데이트를 마치게 된다.

[0097] 이 과정에서 난수 값의 변경이 발생한 노드로부터 상기 루트노드에 이르는 상위 노드들은 변경된 새로운 난수 값에 의해 변경된 해쉬 값들 역시 업데이트 하게 된다. 즉, 위와 같이 d=1,...,c에 대해 $H(r'_{\lfloor i/2^d \rfloor})$의 값을 새로 계산된 $H(r_{\lfloor i/2^d \rfloor} \| H(r'_{2\lfloor i/2^d \rfloor}) \| H(r'_{2\lfloor i/2^d \rfloor+1}))$으로 변경한다.

[0098] 도 9 내지 도 14는 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템을 설명하기 위해 도시한 블록도이다.

[0099] 도 9를 참조하면, 본 발명의 일 실시예에 따른 검증 가능한 데이터 관리 시스템(900)은 데이터 설정부(910), 데이터 업로드부(920), 데이터 검색부(930), 데이터 검증부(940), 및 데이터 업데이트부(950)를 포함할 수 있다.

[0100] 먼저, 상기 데이터 설정부(910)는 도 10에 도시된 바와 같이 제1 데이터 설정부(1010), 해쉬이진트리 구성부(1020), 및 제2 데이터 설정부(1030)를 포함할 수 있다.

[0101] 상기 제1 데이터 설정부(1010)는 상기 클라이언트에 배치된다. 상기 제1 데이터 설정부(1010)는 시큐리티 파라미터(security parameter)와 저장 가능한 데이터의 최대 개수를 설정하고, 상기 설정된 시큐리티 파라미터에 기초하여 서명키, 확인키 및 의사난수함수를 생성한다. 상기 제1 데이터 설정부(1010)는 상기 생성된 데이터 중에서 상기 확인키 및 상기 의사난수함수를 상기 서버에 전달한다.

[0102] 상기 해쉬이진트리 구성부(1020)는 상기 서버 및 상기 클라이언트에 배치된다. 상기 해쉬이진트리 구성부(1020)는 상기 의사난수함수 및 해쉬함수를 이용하여 상기 설정된 최대 개수로 이루어진 해쉬이진트리를 구성한

다.

[0103] 이를 위해, 상기 해쉬이진트리 구성부(1020)는 임의의 난수 값, 및 2개의 자식노드의 값을 각각 파라미터로 하는 2개의 해쉬함수를 이용하여 최하위노드를 제외한 각 노드의 값을 산출하고, 상기 최하위노드를 포함하여 상기 각 노드를 부모노드와 자식노드의 관계를 이용해 상기 루트노드로부터 상기 최하위노드까지 상관성을 가지도록 연결함으로써 상기 해쉬이진트리를 구성할 수 있다.

[0104] 상기 제2 데이터 설정부(1030)는 상기 클라이언트에 배치된다. 상기 제2 데이터 설정부(1030)는 상기 서버로부터 상기 해쉬이진트리의 루트노드의 값을 전송받아 자신이 계산한 루트노드의 값과 비교하고, 상기 비교 결과 일치하면, 상기 루트노드의 값을 저장한다.

[0105] 다음으로, 상기 데이터 업로드부(920)는 상기 클라이언트가 자신의 i번째 데이터를 서버에 저장하고자 하는 경우에 수행될 수 있다. 이러한 데이터 업로드부(920)는 제1 데이터 업로드부(1110) 및 제2 데이터 업로드부(1120)를 포함할 수 있다.

[0106] 상기 제1 데이터 업로드부(1110)는 상기 클라이언트에 배치될 수 있다. 상기 제1 데이터 업로드부(1110)는 상기 의사난수함수를 이용하여 난수 값을 계산하고, 상기 난수 값과 상기 i번째 데이터의 연산 결과 데이터에 대한 서명을 상기 서명키를 이용하여 생성한 후 상기 서버에 전송할 수 있다.

[0107] 상기 제2 데이터 업로드부(120)는 상기 서버에 배치될 수 있다. 상기 제2 데이터 업로드부(120)는 상기 연산 결과 데이터에 대해 상기 서명이 정당한지를 상기 확인키를 이용하여 검증하고, 상기 검증 결과 상기 서명이 정당하면, 상기 연산 결과 데이터 및 상기 서명을 상기 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장할 수 있다.

[0108] 다음으로, 상기 데이터 검색부(930)는 상기 클라이언트가 i번째 데이터를 검색하고자 하는 경우에 수행될 수 있다. 이러한 데이터 검색부(930)는 수신부(1210), 탐색부(1220), 및 전송부(1230)를 포함할 수 있다.

[0109] 상기 수신부(1210)는 상기 서버에 배치될 수 있다. 상기 수신부(1210)는 상기 i번째에 해당하는 인덱스를 상기 클라이언트로부터 전송받을 수 있다.

[0110] 상기 탐색부(1220)는 상기 서버에 배치될 수 있다. 상기 탐색부(1220)는 상기 수신부(1210)로부터 전송받은 상기 인덱스를 이용하여, 상기 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색할 수 있다. 또한, 상기 탐색부(1220)는 상기 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색할 수 있다.

[0111] 상기 전송부(1230)는 상기 탐색된 난수 값 및 해쉬 값을 상기 연산 결과 데이터 및 상기 서명과 함께 상기 클라이언트에게 검증 값으로 전송할 수 있다.

[0112] 다음으로, 상기 데이터 검증부(940)는 상기 클라이언트가 i번째 데이터의 정당성을 검증하고자 하는 경우에 수행될 수 있다. 이러한 데이터 검증부(940)는 제1 검증부(1310), 제2 검증부(1320), 및 검증 완료부(1330)를 포함할 수 있다.

[0113] 상기 제1 검증부(1310)는 상기 클라이언트에 배치될 수 있다. 상기 제1 검증부(1310)는 상기 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증할 수 있다.

[0114] 상기 제2 검증부(1320)는 상기 클라이언트에 배치될 수 있다. 상기 제2 검증부(1320)는 상기 제1 검증 결과 정당한 서명인 경우, 상기 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증할 수 있다.

[0115] 이를 위해, 상기 제2 검증부(1320)는 상기 해쉬이진트리의 각 노드에 저장되어 있는 값과 상기 서버로부터 수신한 각 노드의 값을 제1 비교하여 동일한지 여부를 확인할 수 있다. 또한, 상기 제2 검증부(1320)는 상기 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교하여 동일한지 여부를 확인할 수 있다.

[0116] 상기 제2 검증부(1320)는 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증할 수 있다. 즉, 상기 제2 검증부(1320)는 상기 제1 및 제2 비교 결과 모두 동일한 것으로 확인되면, 상기 난수 값에 대해 정당한 것으로 검증 완료할 수 있으며, 상기 제1 및 제2 비교 결과 하나라도 동일하지 않으면 상기 난수 값에 대해 정당하지 않은 것으로 검증 완료할 수 있다.

[0117] 상기 검증 완료부(1330)는 상기 클라이언트에 배치될 수 있다. 상기 검증 완료부(1330)는 상기 제2 검증 결과

상기 난수의 값이 정당한 경우, 상기 i번째 데이터의 정당성을 인정할 수 있다. 즉, 상기 검증 완료부(1330)는 상기 i번째 데이터가 정당한 것으로 최종 검증 완료할 수 있다.

[0118] 다음으로, 상기 데이터 업데이트부(950)는 상기 클라이언트가 i번째 데이터를 업데이트하고자 하는 경우에 수행될 수 있다. 이러한 데이터 업데이트부(950)는 제1 데이터 업데이트부(1410), 및 제2 데이터 업데이트부(1420)를 포함할 수 있다.

[0119] 상기 제1 데이터 업데이트부(1410)는 상기 클라이언트에 배치될 수 있다. 상기 제1 데이터 업데이트부(1410)는 상기 i번째 데이터의 검증 결과 정당성이 인정되면, 상기 의사난수함수를 이용하여 새로운 난수 값을 계산하고, 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성할 수 있다.

[0120] 상기 제1 데이터 업데이트부(1410)는 상기 i번째 노드에 저장되어 있는 난수 값을 상기 새로운 난수 값으로 변경하고, 상기 검증값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산할 수 있다.

[0121] 상기 제1 데이터 업데이트부(1410)는 상기 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 상기 서명, 및 상기 계산된 루트노드의 값을 상기 서버에 전송할 수 있다.

[0122] 상기 제2 데이터 업데이트부(1420)는 상기 서버에 배치될 수 있다. 상기 제2 데이터 업데이트부(1420)는 상기 검증 값에 포함된 각 노드들의 값을 상기 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산할 수 있다.

[0123] 상기 제2 데이터 업데이트부(1420)는 상기 자신이 보유한 루트노드의 값과 상기 클라이언트로부터 전송받은 루트노드의 값을 비교하고, 상기 비교 결과 두 값이 동일하면, 상기 i번째 데이터에 대한 업데이트를 완료할 수 있다.

[0124] 본 발명의 실시예들은 다양한 컴퓨터로 구현되는 동작을 수행하기 위한 프로그램 명령을 포함하는 컴퓨터 판독 가능 매체를 포함한다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 로컬 데이터 파일, 로컬 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체는 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체, CD-ROM, DVD와 같은 광기록 매체, 플롭티컬 디스크와 같은 자기-광 매체, 및 롬, 램, 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다.

[0125] 지금까지 본 발명에 따른 구체적인 실시예에 관하여 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서는 여러 가지 변형이 가능함은 물론이다. 그러므로, 본 발명의 범위는 설명된 실시예에 국한되어 정해져서는 안 되며, 후술하는 특허 청구의 범위뿐 아니라 이 특허 청구의 범위와 균등한 것들에 의해 정해져야 한다.

[0126] 이상과 같이 본 발명은 비록 한정된 실시예와 도면에 의해 설명되었으나, 본 발명은 상기의 실시예에 한정되는 것은 아니며, 이는 본 발명이 속하는 분야에서 통상의 지식을 가진 자라면 이러한 기재로부터 다양한 수정 및 변형이 가능하다. 따라서, 본 발명 사상은 아래에 기재된 특허청구범위에 의해서만 파악되어야 하고, 이의 균등 또는 등가적 변형 모두는 본 발명 사상의 범주에 속한다고 할 것이다.

부호의 설명

[0127] 910: 데이터 설정부

920: 데이터 업로드부

930: 데이터 검색부

940: 데이터 검증부

950: 데이터 업데이트부

1010: 제1 데이터 설정부

1020: 해쉬이진트리 구성부

1030: 제2 데이터 설정부

1110: 제1 데이터 업로드부

1120: 제2 데이터 업로드부

1210: 수신부

1220: 탐색부

1230: 전송부

1310: 제1 검증부

1320: 제2 검증부

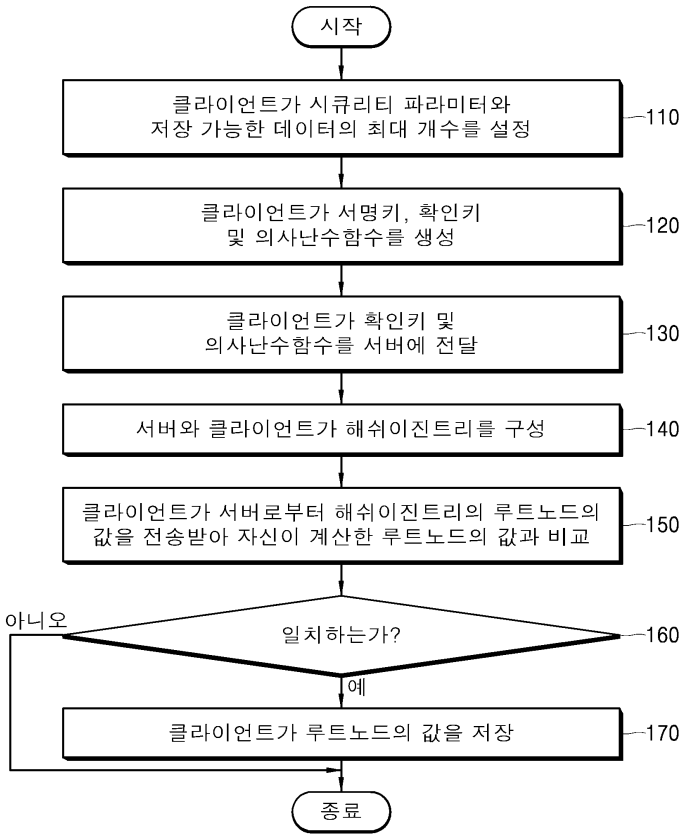
1330: 검증 완료부

1410: 제1 데이터 업데이트부

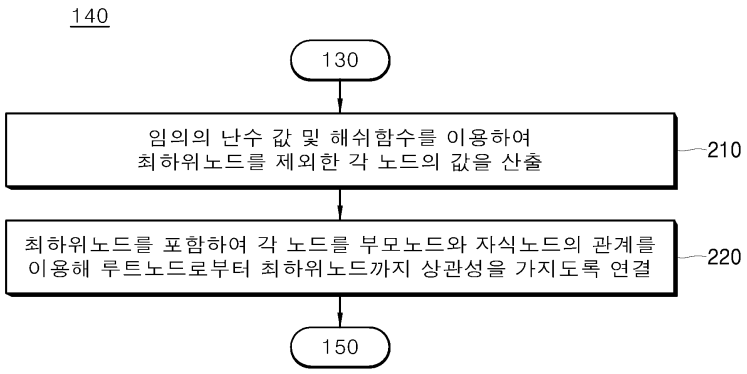
1420: 제2 데이터 업데이트부

도면

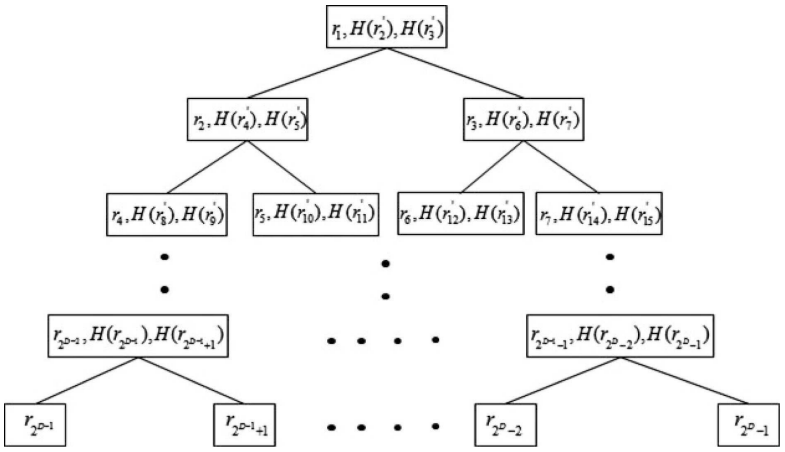
도면1



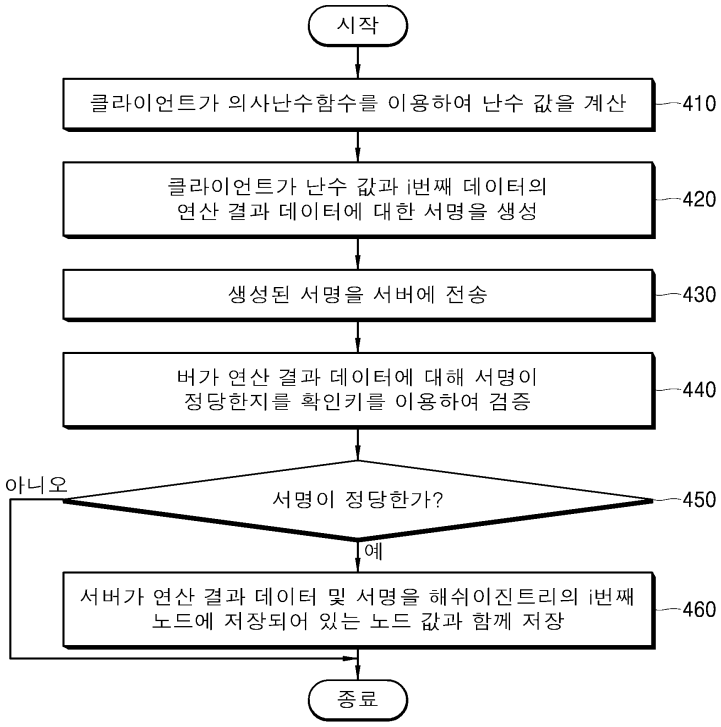
도면2


140
130
임의의 난수 값 및 해쉬함수를 이용하여
최하위노드를 제외한 각 노드의 값을 산출
210
최하위노드를 포함하여 각 노드를 부모노드와 자식노드의 관계를
이용해 루트노드로부터 최하위노드까지 상관성을 가지도록 연결
220
150

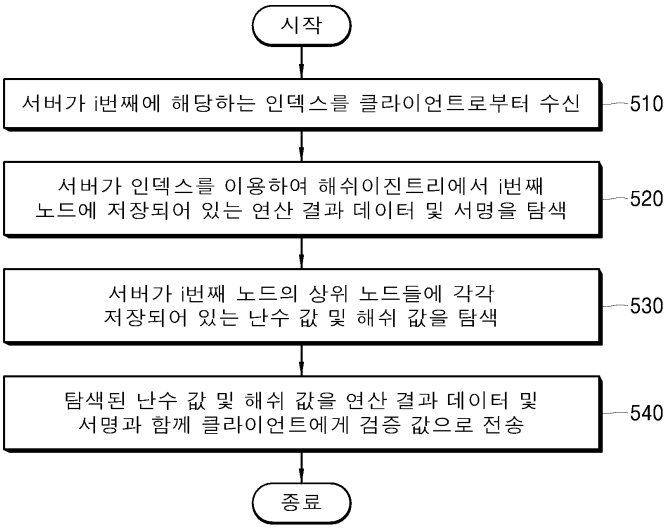
도면3



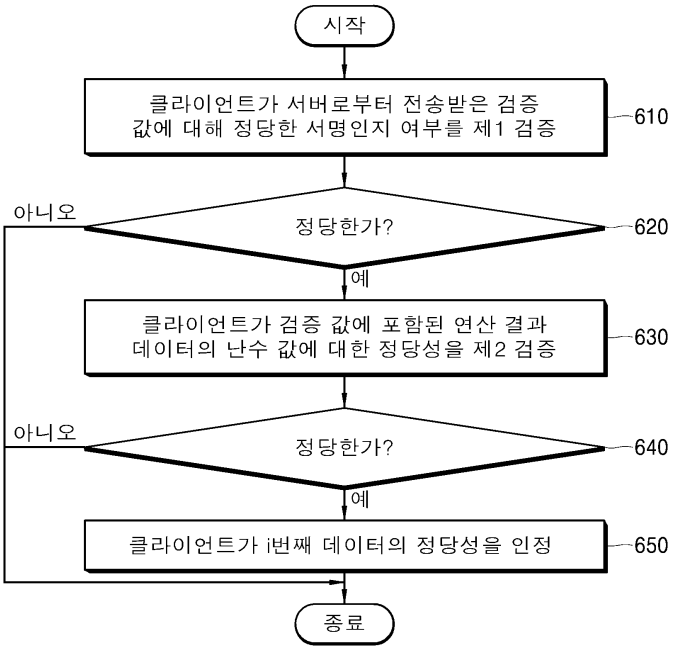
도면4


시작
클라이언트가 의사난수함수를 이용하여 난수 값을 계산
410
클라이언트가 난수 값과 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성
420
생성된 서명을 서버에 전송
430
버가 연산 결과 데이터에 대해 서명이 정당한지를 확인키를 이용하여 검증
440
서명이 정당한가?
450
아니오
예
서버가 연산 결과 데이터 및 서명을 해쉬이진트리의 i번째 노드에 저장되어 있는 노드 값과 함께 저장
460
종료

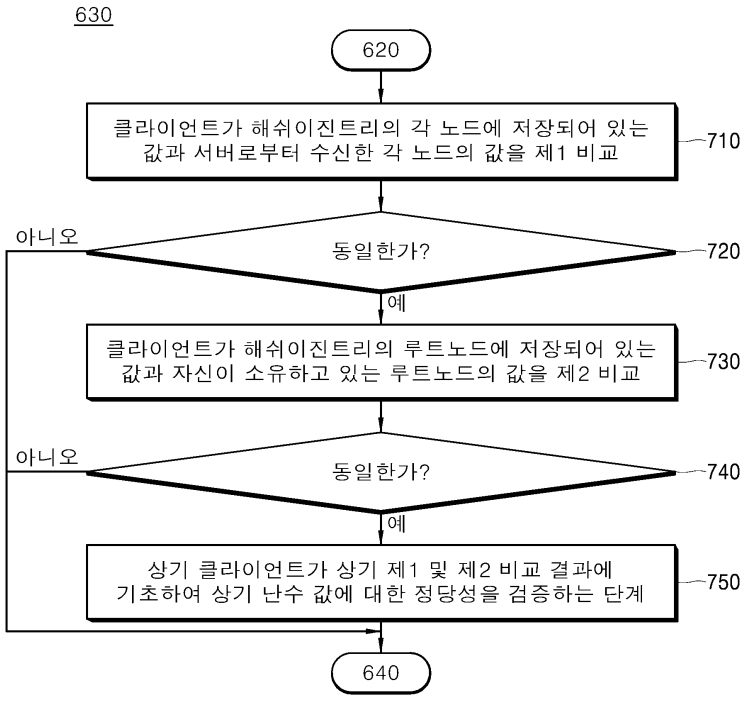
도면5


시작
서버가 i번째에 해당하는 인덱스를 클라이언트로부터 수신
510
서버가 인덱스를 이용하여 해쉬이진트리에서 i번째 노드에 저장되어 있는 연산 결과 데이터 및 서명을 탐색
520
서버가 i번째 노드의 상위 노드들에 각각 저장되어 있는 난수 값 및 해쉬 값을 탐색
530
탐색된 난수 값 및 해쉬 값을 연산 결과 데이터 및 서명과 함께 클라이언트에게 검증 값으로 전송
540
종료

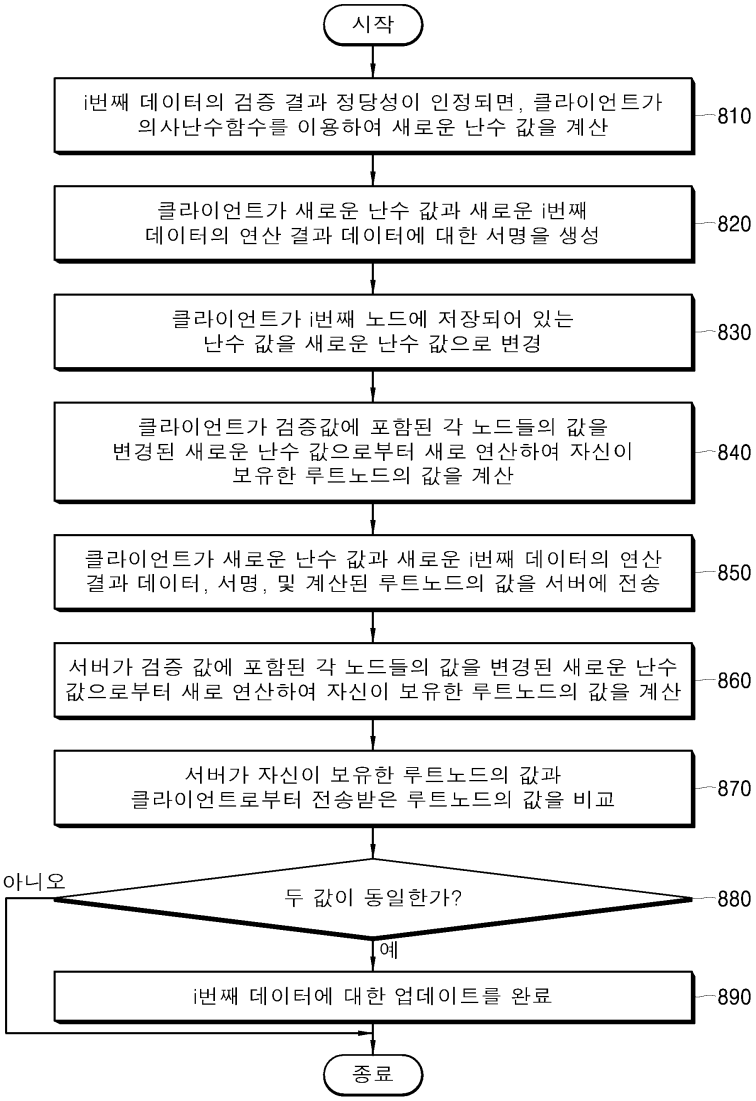
도면6


시작
클라이언트가 서버로부터 전송받은 검증 값에 대해 정당한 서명인지 여부를 제1 검증
610
정당한가?
620
아니오
예
클라이언트가 검증 값에 포함된 연산 결과 데이터의 난수 값에 대한 정당성을 제2 검증
630
정당한가?
640
아니오
예
클라이언트가 i번째 데이터의 정당성을 인정
650
종료

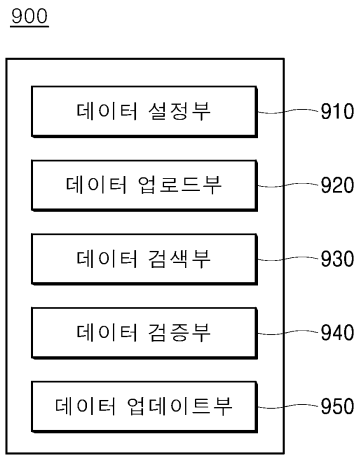
도면7


630
620
클라이언트가 해쉬이진트리의 각 노드에 저장되어 있는 값과 서버로부터 수신한 각 노드의 값을 제1 비교
710
동일한가?
720
아니오
예
클라이언트가 해쉬이진트리의 루트노드에 저장되어 있는 값과 자신이 소유하고 있는 루트노드의 값을 제2 비교
730
동일한가?
740
아니오
예
상기 클라이언트가 상기 제1 및 제2 비교 결과에 기초하여 상기 난수 값에 대한 정당성을 검증하는 단계
750
640

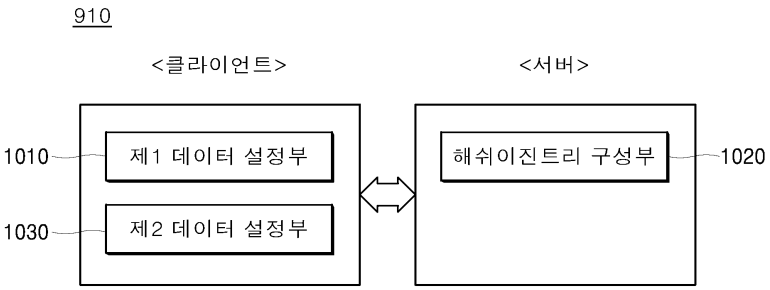
도면8


시작
i번째 데이터의 검증 결과 정당성이 인정되면, 클라이언트가 의사난수함수를 이용하여 새로운 난수 값을 계산
810
클라이언트가 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터에 대한 서명을 생성
820
클라이언트가 i번째 노드에 저장되어 있는 난수 값을 새로운 난수 값으로 변경
830
클라이언트가 검증값에 포함된 각 노드들의 값을 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산
840
클라이언트가 새로운 난수 값과 새로운 i번째 데이터의 연산 결과 데이터, 서명, 및 계산된 루트노드의 값을 서버에 전송
850
서버가 검증 값에 포함된 각 노드들의 값을 변경된 새로운 난수 값으로부터 새로 연산하여 자신이 보유한 루트노드의 값을 계산
860
서버가 자신이 보유한 루트노드의 값과 클라이언트로부터 전송받은 루트노드의 값을 비교
870
아니오
두 값이 동일한가?
880
예
i번째 데이터에 대한 업데이트를 완료
890
종료

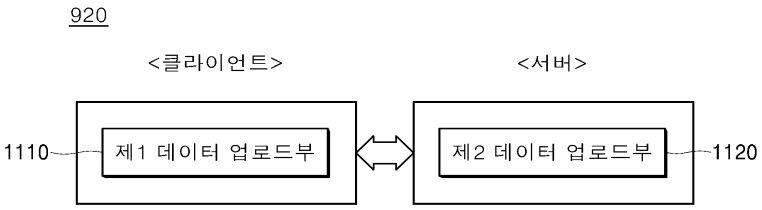
도면9


900
데이터 설정부
910
데이터 업로드부
920
데이터 검색부
930
데이터 검증부
940
데이터 업데이트부
950

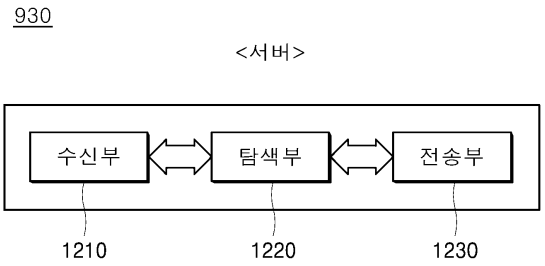
도면10



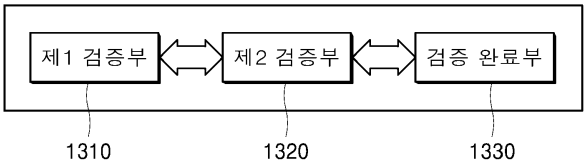
도면11



도면12



도면13



도면14

