JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 6, pp. 1581-1588, Jun. 2024

Check for updates

블록체인 기반 클라우드 데이터의 안전한 저장에 관한 연구

단 관[1] · 손 이 심[2] · 고 금 명[2] · 정 회 경[3*]
[1]웨이팡과학기술대학교 컴퓨터공학과 교수
[2]배재대학교 컴퓨터공학과 박사과정
[3]배재대학교 컴퓨터공학과 교수

A Study on Secure Storage of Cloud Data Based on Blockchain

Juan Tan[1] · Yixin Sun[2] · Jin-ming Gao[2] · Hoe-kyung Jung[3*]

[1]Professor, Department of Computer Engineering, Weifang University of Science and Technology, Shandong 262799, China
[2]Ph.D. Course, Department of Computer Engineering, Pai Chai University, Daejeon 35345, Korea
[3]Professor, Department of Computer Engineering, Pai Chai University, Daejeon 35345, Korea

[요 약]

블록체인은 변조 방지 및 탈중앙화의 특성을 가지고 있어 데이터의 안전한 저장을 보장할 수 있지만 클라우드 데이터 스토리지에 직접 적용하기에는 몇 가지 단점이 있다. 한편으로는 블록의 용량이 작기 때문에 대규모 클라우드 데이터의 저장 요구를 충족시킬 수 없다. 반면 클라우드 데이터 감사에서는 타임스탬프를 기반으로 한 온(on)체인 검색에 오랜 시간이 걸린다. 블록체인 기술과 인터 플래너레이티 파일 시스템(IPFS, Inter Planeraty File System)을 결합하여 클라우드 데이터의 안전한 저장을 보장하기 위해 암호화 기술이 사용된다. 온체인과 오프(off)체인 저장 방법을 결합하면 블록체인 클라우드의 데이터 용량 및 처리량 부족 문제를 해결할 수 있다. 본 논문에서는 타임스탬프를 쿼리 키워드로 하는 클라우드 데이터 감사에서 검색 시간이 긴 문제를 해결하기 위해 블록 헤더에 새로운 필드를 추가하고 블록 본체의 머클(Merkle) 트리를 개선하여 트랜잭션 데이터의 보안을 보장할 뿐만 아니라 체인에서 데이터 검색의 속도와 효율성을 향상시킨다.

[Abstract]

Blockchain, with its tamper-proof and decentralized nature, ensures secure data storage. However, its direct application to cloud data storage faces challenges. The small capacity of blocks cannot accomodate large-scale cloud data, and timestamps based on on-chain retrievals are time-consuming. Cryptography technology is used to ensure the secure storage of cloud data by combining blockchain technology and interplanetary file system. The combination of on-chain and off-chain storage methods can address the data capacity and throughput issues. To enhance retrieval speed in cloud data audits, we added new fields to the block header and improved the Merkle tree in the block body, ensuring transaction security and efficient data retrieval.

색인어 : 머클 트리, 블록체인, 파일 시스템, 클라우드 데이터, 타임스탬프

Keyword : Merkle Tree, Blockchain, File System, Cloud Data, Timestamp

http://dx.doi.org/10.9728/dcs.2024.25.6.1581

Received 23 April 2024; **Revised** 21 May 2024
Accepted 11 June 2024

***Corresponding Author; Hoe-kyung Jung**

Tel: ※ 개인정보 표시제한

E-mail: hkjung@pcu.ac.kr

Ⅰ. 서 론

블록체인 기술은 새로운 유형의 분산 컴퓨팅 모델로서 탈중앙화, 분산 및 비 변조의 특성을 가지므로 전통적인 클라우드 스토리지 서비스에서 단일 장애 지점 및 데이터 프라이버시 유출의 문제를 효과적으로 피할 수 있다[1]-[3].

블록체인 기반 클라우드 데이터 보안 저장 기술은 사용자 데이터를 여러 노드에 저장하고, 각 노드는 독립적이며, 데이터는 분산 원장 형태로 블록체인에 기록된다. 이렇게 하면, 하나의 노드가 실패하거나 공격을 받더라도 다른 노드는 여전히 정상적으로 작동할 수 있고 사용자 데이터는 영향을 받지 않는다[4],[5].

본 논문에서는 인터 플래너레이티 파일 시스템(IPFS, Inter Planeraty File System)을 결합하여 블록체인 기반의 새로운 클라우드 데이터 저장 아키텍처를 설계하고, 온체인과 오프체인 저장 방식의 도움으로 대용량 파일 클라우드 데이터의 블록체인 저장 압력을 해결함과 동시에 클라우드 데이터의 안전한 저장을 보장한다. 또한 데이터 계층 내 머클 트리의 데이터 저장 구조를 개선하여 체인에서 클라우드 데이터를 검색 효율성을 향상시킨다.

Ⅱ. 머클 트리와 인터 플래너레이티 파일 시스템

본 장에서는 데이터의 수집 및 기술 비교분석 등의 내용을 나타낸다.

2-1 블록체인

블록체인은 탈중앙화, 집합유지, 보안 및 신뢰성의 특성을 가지고 있으며 변조할 수 없으며, 그 본질은 데이터를 '블록'에 기록하는 것인데, 이 블록들은 암호화와 타임스탬프에 의해 함께 연결되어 불변의 체인 구조를 형성한다. 탈중앙화는 블록체인 시스템이 중앙 집중화된 기관이나 제3자의 신뢰에 의존하지 않기 때문에 거래 비용을 절감하고 보안을 개선하며 신뢰성을 높일 수 있음을 의미한다. 불변성은 데이터가 블록체인에 한 번 기록되면 수정하거나 삭제할 수 없으므로 데이터의 무결성과 신뢰성을 확보할 수 있음을 의미한다. 블록체인의 구조는 데이터 계층, 네트워크 계층, 인센티브 계층, 계약 계층, 응용 계층을 포함한다. 이 중 데이터 계층, 네트워크 계층, 합의 계층은 블록체인 구조의 핵심으로 저신뢰 환경에서 유통 문제를 해결하기 위한 핵심 기술이다.

2-2 머클트리

머클 트리는 대규모 데이터의 무결성을 빠르게 계산하고 검증하기 위해 트리와 같은 구조를 사용하는 블록체인 구조의 중요한 구성 요소이다[6],[7].

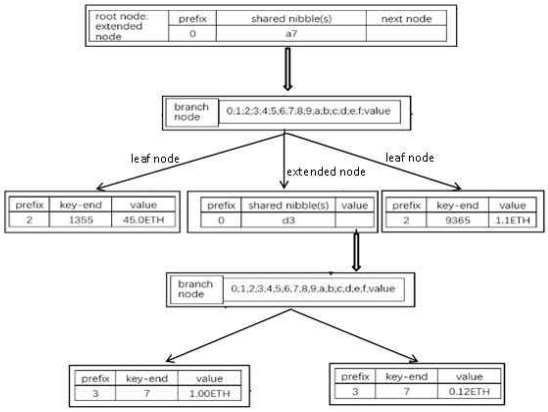

그림 1. MPT 구조의 개략도
Fig. 1. Diagram of the MPT structure

머클 트리는 유효성 검사 효율성이 우수하다. N개의 원소를 가진 머클 트리에서는 log2N개의 해쉬만 있으면 트리에 원소가 있는지 확인할 수 있다.

비트코인에서는 가장 간단한 머클 트리를 블록 헤더에 사용하고 이더리움은 사용자 데이터와 트랜잭션 해시를 관리하기 위해 머클 패트리샤 트리(MPT, Merkle Patricia Tree)를 사용하여 접두사 트리의 장점을 활용하여 머클 트리를 개선했다[8],[9]. 이더리움의 MPT 구조는 그림 1과 같다.

비트코인의 머클 트리와 달리 이더리움의 MPT는 업데이트가 필요한 경우가 많다. MPT의 특징은 다음과 같다. 머클 루트는 노드가 업데이트, 삽입, 삭제된 후 더 빠르게 계산되며, 머클 루트는 데이터가 삽입되는 순서에 관계없이 데이터에만 의존하며, 트리 깊이는 제한되고 조회 속도는 안정적이다[10].

2-3 인터 플래너레이티 파일 시스템

이는 중앙 집중식 서버나 데이터 센터에 의존하지 않고, 대신 상호 연결된 노드 집합을 사용하여 파일을 저장하고 검색하는 P2P 분산 파일 시스템이다. IPFS는 파일을 청크(chunk)로 나누고 내용을 기반으로 고유한 해시를 만드는 방식으로 작동한다. 이것은 데이터 무결성과 검증 가능성을 보장하고 파일을 네트워크 전체에서 공유하고 복제할 수 있도록 한다.

Ⅲ. 블록체인 기반 클라우드 데이터 스토리지 아키텍처

3-1 블록체인 기반 클라우드 데이터 스토리지 아키텍처 (CDSA-BC, Cloud Data Storage Architecture Based on Block Chain)

CDSA-BC의 아키텍처는 관리 모듈과 블록체인 모듈의 두 부분으로 나눌 수 있다. 이 두 모듈은 그림 2와 같은 특징을

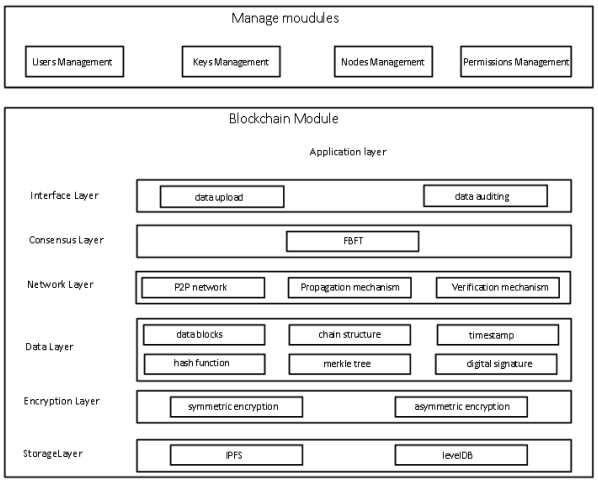

그림 2. CDSA-BC 아키텍처 설계 구성도
Fig. 2. CDSA-BC architecture design diagram

가지고 있다.

관리 모듈은 노드와 사용자를 관리하고 블록 생성 및 노드 합의에 참여하지 않으며 CDSA-BC에 참여하는 노드의 검토, 등록 및 취소, 사용자 계정 관리 및 키 분배 및 권한 관리 등의 작업을 수행한다.

블록체인 모듈은 CDSA-BC에 참여하는 블록체인 노드와 IPFS 노드가 P2P 네트워크로 연결되어 있다. 스토리지층, 암호화층, 데이터층, 네트워크층, 합의층, 인터페이스층, 애플리케이션층의 7개 계층으로 나뉜다.

CDSA-BC는 인센티브 계층과 계약 계층을 제거하고 스토리지 계층과 암호화 계층을 추가한다. 인센티브 계층과 계약 계층을 제거하면 스토리지의 신뢰성과 시스템의 투명성이 저하되지만 CDSA-BC 아키텍처에서는 이러한 단점이 나타나지 않는다. 인센티브 계층의 메커니즘은 블록체인을 유지하는 개인 또는 조직 구조의 노드에 보상하도록 설계되었지만 이러한 노드는 데이터를 저장하기 위해 반드시 블록체인을 사용할 필요가 없으며 이는 주로 퍼블릭 체인에 적용된다[11]. CDSA-BC에서 노드는 컨소시엄 체인 형태로 합의에 도달하고 스토리지 요구가 있는 노드는 합의에 참여하기 전에 검토해야 하므로 인센티브 메커니즘은 이 아키텍처에 적합하지 않다. 계약 계층의 목적은 거래에서 중개자가 필요하다는 문제를 해결하여 거래 비용을 줄이는 것이므로 CDSA-BC에서는 계약 계층이 필요하지 않다. 반대로 CDSA-BC가 제안한 아키텍처에 추가된 스토리지 계층과 암호화 계층은 보안을 더욱 향상시키고 클라우드 데이터의 보안을 더 잘 보호할 수 있다.

3-2 클라우드 데이터에 대한 보안 스토리지 방법

CDSA-BC는 사용자, 블록체인 노드, IPFS 노드 등 크게 3개의 개체로 구성된다. 클라우드 데이터 전송 및 저장 과정에서 사용자는 대칭 또는 비대칭 암호화 알고리즘을 사용하여 클라우드 데이터의 기밀성을 보장하기 위해 원본 클라우드 파일을 암호화한다. 또한 사용자는 암호문에 서명하기 위해 자신만의 고유한 비대칭 암호화 개인 키를 사용해야 한다. 클라우드 데이터를 감사해야 할 때는 먼저 블록체인 노드에 쿼리 요청을 보내 해당 클라우드 데이터 해시 태그를 얻은 다음 Htag 기반의 IPFS 노드에서 암호화된 클라우드 데이터 암호문을 얻은 다음 키를 사용하여 암호문을 복호화한다.

블록체인 노드는 클라우드 데이터의 해시태그를 보유하고 사용자의 업로드 및 감사 요청에 대응하는 역할을 한다. 사용자가 업로드한 Htag를 지속적으로 저장하고 타임스탬프 키워드에 따라 블록체인에서 조회하여 해당 Htag 목록을 반환한다. 노드간 데이터의 분산 저장은 컨소시엄 체인 형태로 이루어지며, PBFT(Practical Byzantine Fault Tolerant) 합의 알고리즘은 CDSA-BC에서 데이터의 일관성을 보장한다. IPFS 노드는 암호화된 원본 클라우드 데이터 파일을 저장하고 사용자의 업로드 요청에 따라 서버에 파일을 저장한 후 다른 노드와 공유한다. 클라우드 데이터 감사 시 IPFS 노드는 사용자가 블록체인 노드에서 조회한 Htag를 기반으로 암호화된 DC 클라우드를 반환한다. 그림 3은 CDSA-BC 전체에 대한 클라우드 데이터의 보안 저장 방법을 보여주는데, 클라우드 데이터 업로드와 클라우드 데이터 감사의 두 가지 주요 부분으로 구성된다.

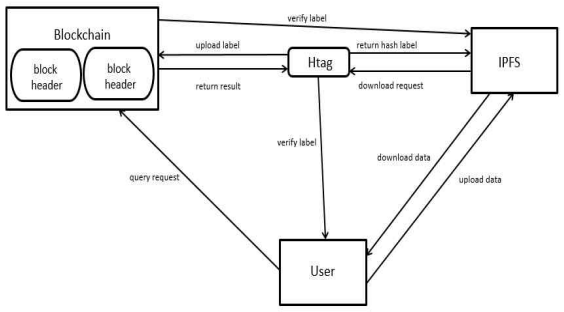

그림 3. CDSA-BC 클라우드 데이터 보안 저장 방법
Fig. 3. CDSA-BC cloud data security storage method

Ⅳ. 블록체인의 머클 트리 개선

4-1 머클 이진 탐색 트리(MBST, Merkle Binary Search Tree)

MBST에서는 전통적인 머클 트리의 비효율적인 검색 문제를 해결하기 위해 각 노드에 타임스탬프 필드를 추가하였으며, 노드의 구조는 그림 4와 같다. 모든 노드의 데이터는 키-값 쌍의 형태로 비관계형 데이터베이스 LevelDB에 저장되며, 여기서 각 키는 노드의 해시값에 해당하며, 각 값은 노드의 데이터에 해당한다.

루트 노드에는 분기 노드의 해시 값을 다음 레벨에 저장하는 하나의 값 필드만 포함된다.

분기 노드는 왼쪽 및 오른쪽 자식 노드의 해시 값과 최대 및 최소 클라우드 데이터 타임스탬프를 포함한다. leaf 노드

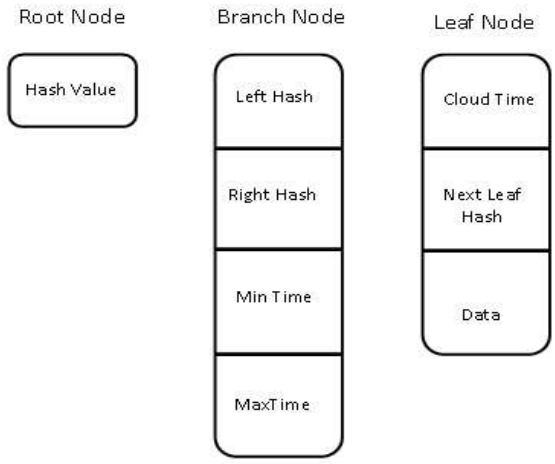

그림 4. MBST 노드 구조의 개요도
Fig. 4. Diagram of MBST node structure

는 클라우드 데이터 시간, next leaf 노드 해시, 클라우드 데이터의 세 부분으로 구성된 가변길이 노드이다. 이 중 클라우드 데이터 시간은 leaf 노드가 저장한 클라우드 데이터의 생성 시간으로 클라우드 데이터의 검색 속도를 향상시킬 수 있다. 이 시나리오에서 모든 노드는 키 값 쌍의 형태로 데이터베이스에 저장되며 노드의 해시 값을 사용하여 데이터베이스에서 다른 노드의 전체 데이터를 얻을 수 있다.찾을 때 루트 노드에서 시작하여 분기 노드의 최대 및 최소 시간을 하나씩 비교하고 마지막으로 잎 노드로 이동하여 클라우드 데이터를 얻는다. 또한, 클라우드 데이터 시간은 유닉스 타임스탬프를 이용하여 정규화한다. 그림 5는 기본 방식의 머클 트리 구조의 간단한 예를 보여주는데, 여기서 회색 부분은 검색 경로를 나타낸다.

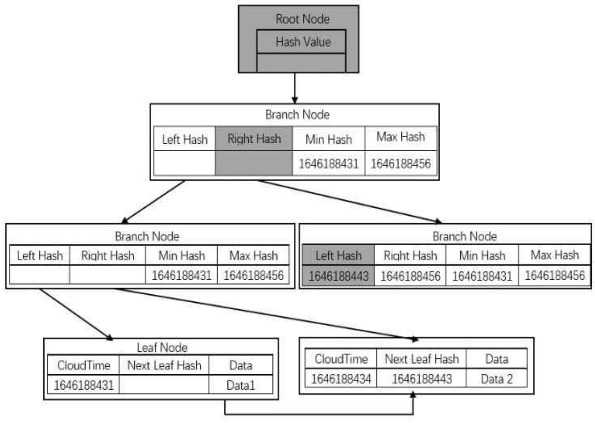

그림 5. 머클 트리의 간단한 예
Fig. 5. A simple example of a Merkle tree

1) MBST 단일점 시간 질의 알고리즘

유닉스 타임스탬프 T와 머클 루트가 주어지면 알고리즘은 지정된 타임스탬프에서 트랜잭션 목록과 머클 경로를 반환한다.

Algorithm	MBST range time query algorithm
input	Unix timestamp Ts and Te, Merkle root Root
output	transaction list Llist, Merkle path Plist
1	node = get Value(root)
2	If T≥node.Min Time and T < node.Max Time then
3	Left Node = get Value(node.leftHash)
4	Right Node = get Value(node.rightHash)
5	Plist Add(node)
6	Replace node with left Node and right Node, Recursively proceed for 2-5 steps until the first leaf Node greater than or equal to Ts is found leaf Node
7	If leaf Node.Cloud Time≥Ts then
8	Next leaf Node = get value(leafNode.nextleafHash)
9	while next leaf node.CloudTime ≤Te
10	Plist Add(leaf Node)
11	Llist Add(leaf Node.Data)
12	Next leaf Node = get Value(nextleafNode.nextleafHash)

2) MBST의 시간 질의 알고리즘

시작 타임스탬프 Ts, 종료 타임스탬프 Te 및 머클의 루트가 주어지면, 이 알고리즘은 [Ts,Te] 기간 및 해당 머클 경로의 트랜잭션 목록을 반환한다.

Algorithm	MBST single point time query algorithm
input	Unix timestamp T, Merkle root Root
output	transaction list Llist, Merkle path Plist
1	node = getValue(root);// getValue 0 Represents getting data from a database
2	If T≥node.Min Time and T < node.Max Time then
3	Left Node = get Value(node.leftHash)
4	Right Node = get Value(node.rightHash)
5	Plist Add(node)
6	Replace node with left Node and right Node Recursively proceed for 2-5 steps until they are leaf Nodes
7	If leaf Node.Cloud Time ==T then
8	Plist Add(leaf Node)
9	Llist.Add (leafNode.Data)
10	return Llist, Plist

4-2 머클 사전 트리(MTT, Merkle Dictionary Tree)

MTT에서는 사전 트리의 특성을 결합하여 온체인 검색 속도를 더욱 향상시킨다. LDSA-BC에서는 각 블록 아래의 모든 클라우드 데이터 목록이 유닉스 타임스탬프를 표준 사양으로 하여 생성된다[12]. 클라우드 데이터 타임스탬프 사이

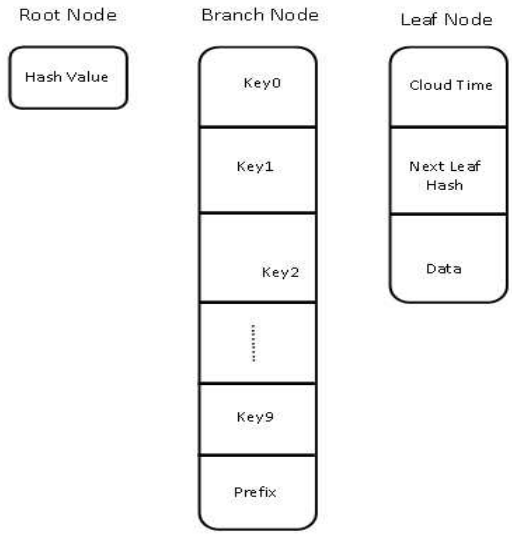

그림 6. MTT 구조도
Fig. 6. MTT structure diagram

의 간격이 충분히 작으면 동일한 공통 접두사를 갖게 된다. 데이터 쿼리 속도를 향상시키기 위해 사전 트리는 문자열의 공통 접두사를 사용하여 트리의 높이를 압축한다. 그림 6에 MTT 구조도를 보인다.

MTT의 검색 키워드는 10진수 인코딩 형식의 유닉스 타임스탬프 문자열이므로 각 분기 노드에는 다음 노드 해시를 저장하는 10개의 필드가 포함되어 있다. 분기 노드에는 트리의 높이를 압축하는 프리픽스 필드도 포함되어 있다. 이 시나리오에서 MTT의 구성은 상향식이며, 먼저 유지해야 하는 블록 클라우드 데이터 목록을 정리하고 클라우드 데이터 생성 시간이 동일한 클라우드 데이터를 동일한 리프 노드에 통합한 다음 각 리프 노드의 타임스탬프에 따라 분기 노드를 생성한다. 그림 7은 MTT 구조의 한 예를 보여준다.

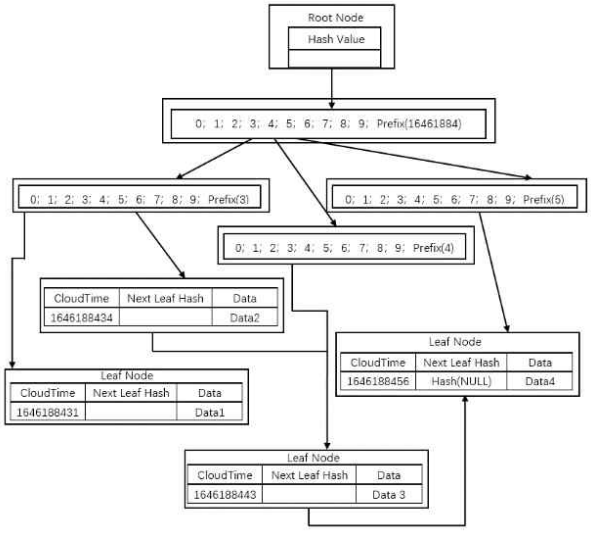

그림 7. MTT 구조 예
Fig. 7. MTT structure example

1) MTT의 단일 시점 질의 알고리즘

유닉스 타임스탬프 T와 머클 루트가 주어지면 알고리즘은 지정된 타임스탬프에서 트랜잭션 목록과 머클 경로를 반환한다.

Algorithm	MTT single point time query algorithm
input	Unix timestamp T, Merkle root Root
output	Transaction list Llist, Merkle path Plist
1	node = get Value(root)
2	for key in node keys
3	If key not equal NULL and compare Prefix(T,node. prefix, key) then
4	Plist.Add(root)
5:	Next Node=get Value(key.Hash)
6:	Replace node with next Node, Recursively proceed in 2-5 steps until the leaf Node leafNode
7:	if leafNode.Cloud Time = T then
8:	Plist.Add (leaf Node)
9:	Llist.Add(leaf Node.Data)
10:	return Llist,Plist

2) 범위 시간 쿼리 알고리즘

알고리즘은 시작 타임스탬프 Ts, 종료 타임스탬프 Te 및 머클 루트 노드 Root를 제공하며, 알고리즘은 시간 범위 [Ts,Te]의 트랜잭션 목록과 해당 머클 경로를 반환한다.

Algorithm	MTT range time query algorithm
input	Unix timestamp Ts and Te, Merkle root Root
output	Transaction list Llist, Merkle path Plist
1	node = get Value(root)
2	for key in node keys
3	If key not equal NULL and compare Prefix(T,node. prefix, key) then
4	Plist.Add(root)
5	Next Node=get Value(key.Hash)
6	Replace node with next Node Recursively proceed in 2-5 steps, Until the first leaf node leafNode greater than or equal to Ts is found
7	if leaf Node.Cloud Time ≥ Ts then
8	Next LeafNode = getValue(leafNode.nextLeafHash):
9	while next Leaf Node.Cloud Time ≤Te
10	Plist.Add (leaf Node)
11	Llist. Add(leaf Node.Data)
12	nextLeafNod= getValue(nextLeafNode.nextLeafHash)
13	return Llist,Plist

4-3 안전성 분석 및 효율성 비교

1) 프로그램의 보안 분석 개선

(1) 데이터 무결성

사용자는 다음 알고리즘을 사용하여 트랜잭션 데이터의 무결성을 검증할 수 있다. 검증에 성공하면 전송 과정에서 데이터가 수정되지 않고 쿼리 결과가 완료된다.

Algorithm	MBST and MTT data integrity verification algorithm
input	Merkle path Plist
output	Verify the results Vres
1	root = Plist[0]
2	node = root
3	for index in range (1, Plist.length - 1)
4	res = verifyNodeHash(node, Plist [index], MBST or MTT)
5	If res =False then
6	return Vres = False
7	node = Plist[index]
8	return Vres = True

(2) 데이터 변조 방지

사용자는 다른 노드에 요청을 보내고 블록 헤더 데이터를 다운로드한 후 블록 헤더와 머클 경로의 머클 루트 값을 비교하여 노드가 공격을 받았는지, 데이터가 변조되었는지 여부를 판단할 수 있다.

(3) 위조방지

질의 결과를 암호화함으로써 암호화된 데이터는 개인 키로 서명된다. 시스템은 전송 중 질의 결과의 기밀성과 진위성을 보장할 수 있다.

(4)안전 저장

json 타입의 클라우드 데이터를 업로드하려면 클라이언트 IP, 이메일 주소, 로그인 이름, 클라우드 데이터 시간, 요청 방식, 상태 코드, 바이트 수 등을 기입해야 한다. 업로드 요청이 서버로 전송되면 서버는 데이터를 json 형식으로 그룹화하고 암호화하여 IPFS 노드에 저장한다

2) 효율성 시뮬레이션 평가

BBT(Binary Branch Tree)는 그림 8과 같이 Wang Li가 데이터 샤딩을 기반으로 구축한 2가지 머클 트리이다. MBST, MTT, BBT를 대상으로 시뮬레이션 종합 비교 실험을 수행하였다.

이 실험의 목적은 트리 구축의 시간 비용과 유닉스 타임스탬프 기반 단일 지점 쿼리, 범위 쿼리를 비교하는 것이다. 이를 위해 100,000개의 무작위 클라우드 데이터를 생성하여 개인 클러스터의 IPFS 서버에 업로드한다. 그 후 이 데이터

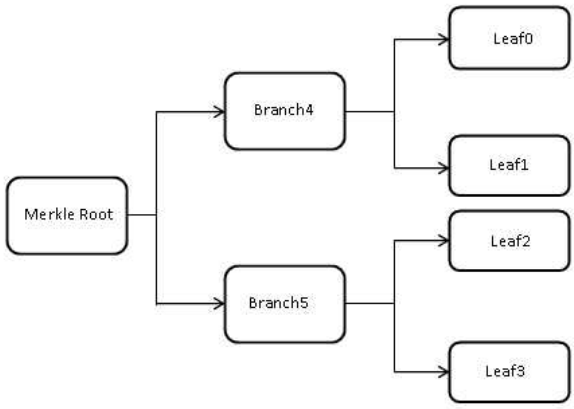

그림 8. BBT 구조의 개략도
Fig. 8. Diagram of the BBT structure

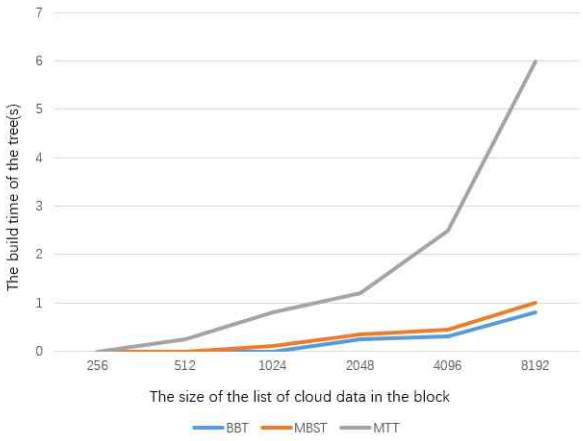

그림 9. BBT, MBST 및 MTT 빌드 시간 비교
Fig. 9. Comparison of BBT, MBST and MTT build times

의 해시 태그를 블록체인에 업로드하여 실험한다.

먼저, 본 논문에서는 트리의 구축 시간과 트랜잭션 목록의 크기 사이의 상관관계를 살펴본다. 그림 9에서와 같이 본 논문에서 제안한 두 가지 개선된 방법은 BBT에 비해 구축 효율이 떨어진다. MBST와 MTT는 구축 과정에서 모든 노드를 데이터베이스에 저장해야 하기 때문에 더 긴 디스크 쓰기 시간이 필요하다. 또한, MTT에서는 모든 클라우드 데이터 생성 시간이 동일한 프리픽스 값이 아닐 경우 다른 두 가지 방식보다 훨씬 많은 수의 트리 노드가 생성된다.

두 번째로 BTT, MBST와 MTT의 쿼리 효율성을 비교한다. 체인에 10,000개의 랜덤 쿼리를 기록하고 평균 쿼리 시간을 비교한다. 그림 10에서 보는 바와 같이 본 논문에서 제안한 두 가지 방식의 단일 포인트 쿼리 시간은 BBT보다 훨씬 짧다. MBST와 MTT는 블록의 모든 트랜잭션을 디스크에서 읽을 필요가 없기 때문이다.

마지막으로, 범위 검색 측면에서도 본 논문에서 제안한 두 가지 방식이 더 나은 쿼리 효율성을 가지고 있다. 구체적으로, 그림 11의 비교는 MTT가 더 낮은 높이와 더 적은 수의 트리 노드를 생성하여 쿼리 시간 측면에서 MBST보다 유리함을

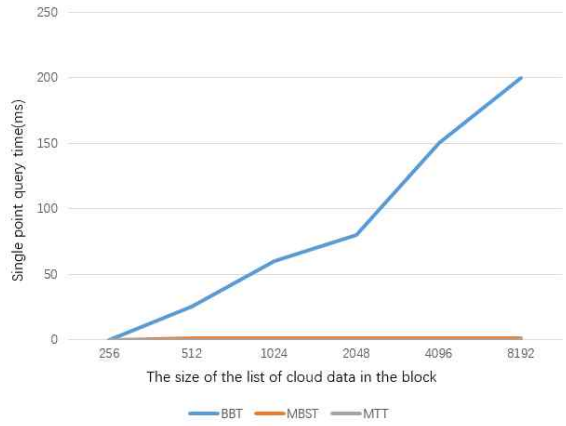

그림 10. BBT, MBST 및 MTT 단일점 질의 시간 비교
Fig. 10. BBT, MBST and MTT single point query time comparison

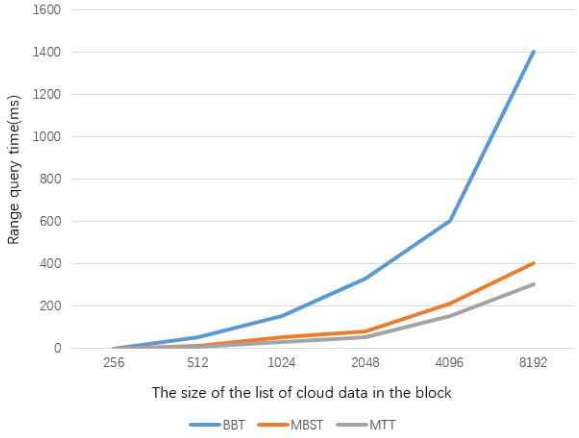

그림 11. BBT, MBST 및 MTT range 쿼리 시간 비교
Fig. 11. Comparison of BBT, MBST and MTT range query times

보여준다. 두 방식의 쿼리 성능은 BBT보다 더 빠르다.

요약하면, 두 가지 개선된 방식인 MBST와 MTT는 BBT보다 트리를 구축하는 데 더 오랜 시간이 걸린다. 그러나 두 방식 모두 효율적인 검색 작업을 지원하기 위해 각 노드에 필드를 추가하여 설계되어 BBT보다 쿼리 비용이 적게든다.

V. 결 론

CDSA-BC는 행성 간 파일 시스템에 클라우드 데이터를 저장하고, 각 클라우드 데이터의 해시 태그는 체인의 블록에 저장된다.

머클 트리의 두 가지 개선인 MBST와 MTT는 블록체인에서 데이터를 검색하는 효율성을 향상시킬 수 있다. 이 설계는 단일 지점 타임스탬프 쿼리 알고리즘과 범위 타임스탬프 쿼리 알고리즘을 사용한다. 시스템 시뮬레이션 실험은 제안된 방식이 온체인 데이터 검색 효율성이 더 빠르다는 것을 보여준다.

감사의 글

이 논문은 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 지역지능화혁신인재양성사업임(IITP-2024-RS-2022-00156334). 본 과제(결과물)는 2024년도 교육부의 재원으로 한국연구재단의 지원을 받아 수행된 지자체-대학 협력기반 지역혁신 사업의 결과입니다(2021RIS-004).

참고문헌

[1] Z. Ming, H. Ying, L. Kui, W. Junkui, and L. Zhi, “Material Inventory Data Security Storage System Based on Blockchain Technology,” *Automation Technology and Application*, pp. 36-39, 2022.

[2] H. Li, D. Han, and M. Tang, “Logistics Chain: A Blockchain-Based Secure Storage Scheme for Logistics Data,” *Mobile Information Systems*, Vol. 2021, 8840399, pp. 1-15, February 2021. https://doi.org/10.1155/2021/8840399

[3] K. Pal and A.-U.-H. Yasar, “Internet of Things and Blockchain Technology in Apparel Manufacturing Supply Chain Data Management,” *Procedia Computer Science*, Vol. 170, pp. 450-457, 2020. https://doi.org/10.1016/j.procs.2020.03.088

[4] Y. Fan, “Research and Application of Log Data Security Storage Based on Blockchain,” *North China University of Technology*, 2022.

[5] C. Han, “Research on User-Side Data Security Storage Mechanism of Smart Grid Based on Blockchain,” *Fujian Institute of Technology,* 2022.

[6] D. Li, D. Han, Z. Zheng, T.-H. Weng, H. Li, H. Liu, ... and K.-C. Li, “MOOCsChain: A Blockchain-Based Secure Storage and Sharing Scheme for MOOCs Learning,” *Computer Standards & Interfaces,* Vol. 81, April 2022. https://doi.org/10.1016/j.csi.2021.103597

[7] M. Mancer, L. S. Terrissa, S. Ayad, and H. Laouz, “A Blockchain-Based Approach to Securing Data in Smart Agriculture,” in *Proceedings of 2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, Biskra, Algeria, pp. 1-5, December 2022.

[8] C. Shunda, “Cloud Data Storage and Access Algorithm Based on Security Verification,” *Journal of Shenyang University of Technology*, 2023.

[9] R. Wang, “Research on Data Security Technology Based on Cloud Storage,” *Procedia Engineering*, Vol. 174, pp. 1340-1355, 2017.
[10] H. Yutong, M. Zhaofeng, and L. Shoushan, “Research and Implementation of Blockchain-Based Data Security Sharing and Controlled Distribution Technology,” *Information Network Security,* pp. 55-63, 2022.
[11] F. Zhengxin, T. Yin, H. Lei, W. Xi, and P. Jing, “Blockchain-Based Secure Sharing of Sensitive Data,” *Information Security Research*, pp. 364-373, 2022.
[12] J. Tan, Research on Secure Storage of Cloud Data Based on Block Chain, Ph.D. Dissertation, Paichai University, Daejeon, June 2023.

단관(Juan Tan)
2005년 : 랴오청 대학 컴퓨터 과학 및 기술학과(공학사)
2010년 : 중국해양대학 컴퓨터 기술학과(공학석사)
2023년 : 배재대학교 컴퓨터공학과(공학박사)
2005년～현 재: 웨이팡과학기술대학교(중국) 컴퓨터공학과 교수
※관심분야 : 인공지능, 빅데이터, 블록체인

손이심(Yixin Sun)
2011년 : 산동공상학원 통계 대학(이학사)
2015년 : 북경항공우주대학 소프트웨어공학과(공학석사)
2023년～현 재: 배재대학교 컴퓨터 공학(박사과정)
※관심분야 : AI, ERP, 디지털 트랜스포메이션

고금명(Jin-Ming Gao)
2018년 : 림기대학 전기 공학 및 자동화(공학사)
2021년 : 샤먼이공대학 전기공학과(공학석사)
2023년～현 재: 배재대학교 컴퓨터 공학(박사과정)
※관심분야 : AI, 에너지 시스템 최적화, 지능형 알고리즘

정회경(Hoe-Kyung Jung)
1985년 : 광운대학교 컴퓨터공학과(공학사)
1987년 : 광운대학교 컴퓨터공학과(공학석사)
1993년 : 광운대학교 컴퓨터공학과(공학박사)
1994년～현 재: 배재대학교 컴퓨터공학과 교수
※관심분야 : 머신러닝, 빅데이터, 임베디드 시스템, U-헬스케어, IoT