효율적 위·변조 탐지 및 무결한 차량 운행 정보의 안정적 질의를 위한 블록체인 기반 분산 데이터 관리 방안 연구

문준오*, 민찬기*, 임종민**, 윤영*
*홍익대학교 컴퓨터공학과 Application Platform Lab
**홍익대학교 경제학과
e-mail: mjo970625@gmail.com
e-mail: alscksrl2010@gmail.com
e-mail: yjm7558@nate.com
e-mail: young.yoon@hongik.ac.kr

Blockchain-based Distributed Database System for Efficient Falsification Detection and Reliable Inquiry of Faultless Automobile Driving Information

Junoh Moon*, Chanki Min*, Jongmin Lim**, Young Yoon*
*Department of Computer Engineering, Hongik University
**Department of Economics, Hongik University

요 약

차량에서 생성되는 데이터의 가치가 상승함에 따라 데이터 소스와 데이터 내용에 대한 보안 위협 또한 증가하고 있다. 데이터 소스인 차량의 경우에는 운행의 안정성을 보장하고자 블록체인을 결합하려는 시도가 있어왔지만, 무결한 차량 운행 데이터 관리 시스템에 대한 이해 부족으로 데이터 위·변조 등 차량 데이터에 대한 사이버 공격에 적절히 대응하지 못하고 있다. 이에 본 논문은 수집된 차량 데이터의 무결성을 보장하고 수집된 데이터에 대한 질의가 가능한 블록체인 기반 데이터베이스 시스템을 제안한다. 본 시스템을 통하여 분산 합의 기반 데이터 무결성 검증, 블록을 구성하는 해시트리의 복제 저장 없이 위·변조된 차량 데이터 검출, 일정 수준의 장애를 허용한 상태 하에서의 질의문 처리 등이 가능해진다. 본 시스템은 높은 공간 효율성과 확장성을 가지며, 수소전기차 공유 업체의 차량 운행 정보를 바탕으로 한 성능 평가 결과 평균적으로 데이터 블록 저장에 4.0 초, 각 블록 검증에 2.4 초, 질의 처리를 위한 합의 과정에 1.3 초가 소요됨을 확인하였다.

1. 서론

통신 기능이 장착된 차량들이 생성하는 운행 정보 등의 데이터에 기반하여 2030 년까지 약 6,000 억 달러 규모의 신규 서비스 시장이 형성될 것이라고 맥킨지는 최근 전망하였다 [1]. 그러나, 차량 데이터의 가치를 노린 보안 공격의 위협 역시 커지고 있다 [1, 2]. 이에 대응하기 위한 블록체인 기반 보안 연구는 데이터 소스인 차량 자체의 안정적 운행에 주로 집중된 편이다 [3]. 반면, 차량 데이터 수집 시스템의 보안체계에 대한 연구 활동은 상대적으로 미진하다 [1, 4].

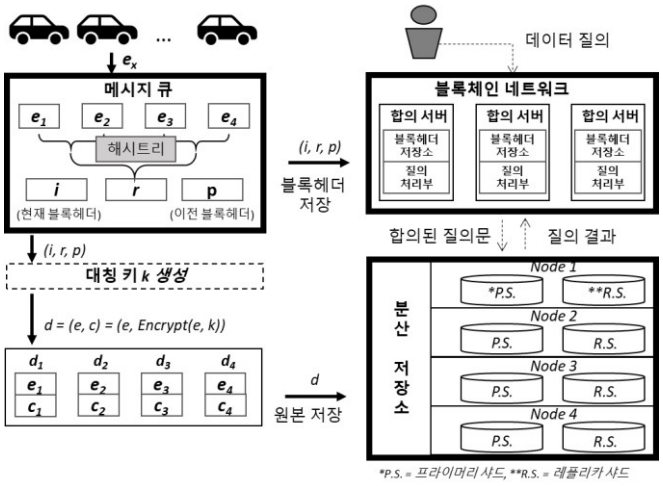
본 논문은 신뢰성 높은 차량 데이터 수집 및 분석 시스템을 위하여 비공개 블록체인을 활용한 차량 데이터의 위·변조 여부 판별법은 물론, 장애 상황에서도 안정적 질의 기능을 제공하는 신개념 데이터베이스 시스템을 제안한다. 본 논문은 다음과 같이 구성된다: 2 장에서는 관련 연구를 다룬다; 3 장에서는 데이터 저장, 검증 및 질의 과정을 위한 시스템 구성을 설명한다; 4 장에서는 시스템의 장애 내성과 위협 모델에 대하여 기술한다; 5 장에서는 시스템의 성능을 평가하고; 6 장에서는 추후 연구 방향과 함께 결론을 내린다.

2. 관련 연구

사토시는 암호화폐 거래의 무결성을 작업 증명에 기반하여 검증할 수 있는 블록체인을 고안하였다 [5]. 사토시의 발표 이래 다양한 방식으로 블록체인을 데이터 저장 및 수집에 활용하려는 시도가 있어왔다. BigchainDB 는 블록체인을 적용한 NoSQL 데이터베이스로 대용량의 데이터 처리를 목적으로 설계되었다 [6]. 그러나 마스터 노드에 대한 의존도가 높은 BigchainDB 는 단일 중단점에 매우 취약하여 분산 블록체인의 이점을 십분 활용했다고 하기에는 무리가 있다. Gaetani 등은 비트코인에서 사용하였던 작업 증

명 방식을 이용하여 클라우드 환경을 위한 데이터베이스를 블록체인 상에서 구현하는 방식을 제안하였다 [7]. 위 방식은 지분 증명 등의 개선된 합의 방식 대신 비효율적인 작업 증명 방식을 사용하고 공개 블록체인 네트워크를 사용하므로 기밀성이 중요한 자료에 대해서는 사용이 어렵다. 또한 데이터 분석을 위한 질의를 기능을 명시하지 않아 데이터 아카이브로 국한된다. Lee 등이 최근 발표한 서비스 제공자 관점에서의 차량 데이터 수집 플랫폼은 보안 및 무결성을 유지하는 방식에 대하여 기술하고 있다 [8]. 그러나 이 논문 역시 제안한 플랫폼 위에서 원하는 데이터를 안정적으로 질의하는 방법에 대해서는 기술하고 있지 않다.

종합적으로, 선행 연구들은 데이터 분석을 위한 질의 기능과 블록체인 기반 검증 기능을 함께 효과적으로 제공하지 못하고 있다고 요약할 수 있다.



(그림 1) 시스템 구성도

3. 시스템 구성

본 논문에서 제안하는 시스템은 크게 데이터를 수집하는 메시징 큐 (queue), 데이터를 검증하는 블록체인 네트워크, 저장부 그리고 질의 처리부로 구성된다. 데이터 저장, 검증 및 질의 시 각 모듈의 수행 방식은 다음과 같다.

3.1. 데이터 저장

메시징 큐는 다수의 차량으로부터 운행 등의 정보 e 를 수집한다. 한 블록이 n 개의 e 로 구성되고, 메시징 큐가 n 개의 e 를 수집하면 그림 1 과 같이 해시트리를 구성한다. 해시트리의 루트는 단말 노드들을 대표한다는 특성이 있으므로, 블록체인 네트워크는 해시트리의 루트 r 만 저장하더라도 n 개의 e 를 한번에 검증할 수 있다. 블록체인 네트워크는 r 을 기반으로 새로운 블록 헤더를 생성한다. 새로운 블록 헤더는 (i, r, p) 3-튜플로 이루어진다. i 는 현재 블록의 번호를, p 는 이전, 즉 $(i-1)$번째 블록 헤더를 해싱한 값이다.

메시징 큐는 (i, r, p)를 시드 (seed)로 삼는 대칭키를 이용하여 각 e 를 cipher c 로 암호화하고, (e, c) 형태의 튜플 d 로 가공하여 저장부에 삽입한다. 저장부는 ElasticSearch[1] 로 구현되어 있으며, (e, c)는 병렬 노드들에 분산 저장되어 공간 효율성 및 확장성을 높게 유지한다. e 외에 c 를 추가로 저장함으로써 추후 위·변조가 일어난 e 를 검출할 수 있는 기술에 대한 구체적인 설명은 다음 장에서 기술한다.

3.2. 데이터 검증

데이터 검증은 설정 가능한 주기마다 모니터링 프로세스 (MP)가 블록체인 네트워크에 각 블록에 대한 검증 요청을 하는 것으로 이루어진다. MP 의 i 번째 블록에 대한 검증을 요청을 수신한 각 블록체인 서버는 Practical Byzantine Fault Tolerance (PBFT) 방식 [9]에 따라 요청이 유효한 것인지 판단한 후 Algorithm 1 을 수행하여 검증을 진행한다. 블록체인 네트워크의 각 서버는 검증 결과를 모니터링 프로세스에 반환하며 위·변조가 발생한 경우 문제를 야기한 d 를 검출한다. MP 는 각 서버가 반환한 검증 결과를 취합한 후 정족수 이상의 일치된 결과를 검증 값으로 확정한다. i 번 블록에 대한 검증이 완료되면 다음 블록 검증을 요청하고 이를 마지막 블록 검증까지 반복한다.

블록 검증 시 재구성한 해시트리의 루트와 블록체인 네트워크에 기 저장되었던 루트가 다를 경우, 위·변조가 발생한 것으로 판단한다. 이 경우, 해시트리 루트만으로는 문제를 야기한 d 를 효율적으로 검출할 수 없다. 위·변조된 d 를 추적하기 위해서는 해시트리 전체가 필요하기 때문이다. 해시트리는 포화 이진트리의 일종이므로 다음의 속성을 지닌다.

해시트리의 모든 노드를 블록체인에 복제하여 저장하는 것은 시스템에 급속한 과부하를 야기할 수 있다. 이를 극복하기 위하여 본 시스템은 cipher c 를 활용하여 새로운 기술을 고안하였다. 데이터의 위·변조가 확인되면 각 서버는 블록을 구성하는 데이터 중 위·변조된 d 를 검출하기 위하여 블록을 구성하는 각 d 에 대하여 Algorithm 2 를 이용하여 무결성 검증 절차를 수

Algorithm 1 Block Integrity Verification

```
function VerifyBlock(block_index)
    block_header ← fetch_from_blockchain(block_index)
    next_block_header ← fetch_from_blockchain(block_index + 1)
    if hash(block_header) ≠ next_block_header.previous_hash then
        return false
    (cipher_list, plain_list) ← fetch_from_database(block_index)
    root′ ← make_merkle_tree(plain_list)
    return block_header.root = root′
```

행한다. 각 d 는 (e, c)로 구성되어 있으므로 e 와 c 모두에 대한 검증이 필요하다. c 의 무결성 판별에는 평문에 접미사(suffix)를 덧붙여 암호화하는 salt/pepper 기법이 활용된다. 외부 공격자는 suffix 와 c 를 암호화하는데 쓰인 대칭키를 탈취하지 않는 이상, c 에 내재된 suffix 를 지니는 c ′ 을 생성할 수 없다. c 의 위·변조 여부는 블록체인 네트워크가 이를 대칭키로 복호

$$n = 2^h \mid \forall n, h \in \mathbb{N} \implies m = w^{h+1} - 1,$$

$$\text{where } m \text{ is the total number of nodes}$$

[1] https://www.elastic.co

화 하였을 때 다른 suffix 값이 반환되는지 여부를 통해 확인 가능하다. c 의 무결성이 확인되었다면, c 를 복호화한 값과 e 를 비교하여 e 의 무결성을 최종 검증한다. 이 과정을 통하여 특정 블록의 위·변조를 발견하였을 시, 문제의 원인이 되는 d 를 검출할 수 있다.

Algorithm 2 Element Integrity Verification

```
function VerifyElement(key, suffix, cipher, entry)
    plain, restored_suffix ← decrypt(cipher, key)
    if restored_suffix ≠ suffix then
        return false
    if plain = entry then
        return true
    else
        return false
```

3.3. 데이터 질의

본 시스템은 단순한 데이터 기록을 위한 아카이브에 그치지 않고, 다양한 분석으로 연계되는 데이터 질의 기능을 수행할 수 있도록 설계되었다. 질의 처리부는 블록체인 네트워크 내부 모듈로 디자인하였다. 질의 처리부는 블록체인의 분산 합의 기능을 활용하여 단일 중단점으로 작용할 수 있는 상황을 방지한다. 즉, 일부 질의 처리부가 장애를 겪더라도 시스템의 안정성을 유지한다. 또한 블록체인 네트워크를 경유하므로 질의문 자체에 대한 무결성을 검증할 수 있다.

본 시스템은 주기적인 블록 검증을 통하여 데이터에 대한 신뢰성을 제공하나 매 질의 결과에 대한 신뢰성 검증은 질의자의 명시적인 요구가 있을 때만 제공된다. 질의 시에 검증을 수행할 경우 참조된 데이터에 대한 전수 검증이 필요하다. 예컨대, 한 테이블에 있는 모든 데이터들인 $\{e_1, e_2, e_3\}$를 검색한다고 가정하자. 그런데 공격자가 테이블 내 e_3 를 악의적인 의도로 삭제할 경우 주기적 검증 프로세스가 이를 감지하기 전까지는 검색 결과로 $\{e_1, e_2\}$가 나올 것이다. $\{e_1, e_2\}$에 대한 무결성은 여전히 유지되더라도 e_3 가 제외되었으므로 위 질의 결과는 잘못되었다. 따라서 질의 결과는 물론 테이블에 대한 전수검사가 필요하며, 이는 오버헤드로 작용한다. 질의 결과에 대한 신뢰성 보증과 질의 응답 시간 증가 간 상충 관계 해소에 대한 후속 연구가 필요하다.

4. 장애와 위협에 대한 내성

4.1. 장애 허용

3 장에서 기술한 절차는 시스템을 구성하는 각 하위시스템이 온전히 작동한다는 조건이 요구된다. 즉, 각 하위 시스템은 일부 시스템 자원에 장애가 발생하더라도 작업을 수행할 수 있도록 장애 허용을 지원해야 한다. 블록체인 네트워크 및 분산 저장소가 장애 상황을 조치하는 방식은 다음과 같다.

블록체인 네트워크는 PBFT 를 이용하여 합의를 이끌어내는데, PBFT 는 시스템이 정지하거나 잘못된 결과를 다른 시스템 자원에 전송하는 등의 장애를 포함하는 비잔틴 장애를 허용할 수 있도록 설계되었다. 구체적으로, 블록체인 네트워크의 2/3 보다 많은 서버가 정상적으로 동작할 시에 나머지 서버가 일으키는 비잔틴 장애를 허용할 수 있다 [10, 9]. 따라서 1/3 미만의 서버가 응답이 없거나, 잘못된 결과를 보내더라도 블록체인 네트워크는 동작이 중단되거나 잘못된 결과를 내지 않고 검증 절차를 성공적으로 수행할 수 있다.

분산 저장소로 사용되는 Elasticsearch 는 원본 데이터를 분산하여 저장한 프라이머리 샤드 (shard)와 이것의 복제본인 레플리카 샤드를 사용하며, 중복 저장을 통한 장애 허용 기능을 갖추고 있다. 프라이머리 샤드의 무결성이 깨진 경우, Elasticsearch 를 관리하는 마스터 노드는 다른 노드에 저장된 레플리카 샤드를 프라이머리 샤드로 변경하여 데이터의 안정성을 유지한다. 마스터 노드가 응답이 없을 경우 마스터 후보 노드들은 선거 (leader election)을 통하여 신규 마스터 노드를 선출하는 방식으로 안정성을 유지한다 [11].

4.2. 위협 모델

본 시스템에 대하여 가능한 위협 및 대응 방안을 소개한다. 첫 번째 위협은 블록체인 네트워크에 대한 공격이다. 블록체인 네트워크는 블록 헤더 저장, 질의 처리 등 시스템에서 핵심적인 역할을 수행하고 있기 때문에 서비스 장애를 목적으로 한 공격이 예상된다. 하지만 4.1 절에서 기술한 바와 같이, 블록체인 네트워크는 PBFT 를 이용하여 합의를 이끌어내기 때문에 $\frac{1}{3}$ 미만의 서버가 장애를 겪더라도 블록체인 네트워크는 안정적이고 신뢰성 있는 서비스를 제공한다.

두 번째 가능한 위협은 분산 저장소에 보관된 e 를 위·변조하는 것이다. 이 공격은 3.2 절에서 기술한 바와 같이, d 로부터 c 를 복호화 한 값과 e 를 비교함으로써 검출할 수 있다.

마지막으로 가능한 위협은 블록체인 네트워크의 일부 서버로부터 대칭키를 탈취하여 i 번 블록 b 내의 $d = (c, e)$를 $d' = (c', e')$로 위·변조하는 경우이다. 이 경우 블록체인 네트워크가 지닌 b 의 루트와 위·변조된 블록 b' 을 통해 재구성한 루트 r' 이 다르므로 검증 프로세스가 b' 안의 위·변조된 d' 을 검출한다. 이때 공격자가 분산 저장소 공격과 동시에 블록체인 네트워크의 일부 서버의 블록 헤더를 위·변조하려는 공격도 시도할 수 있다. 그러나 블록 헤더의 내용은 그 다음 블록에 해싱되어 저장되므로 i 번 이후의 블록은 i 번 블록에 의존적이다. 즉, 검증을 통과하기 위해서는 i 번 이상의 모든 블록 헤더를 위·변조해야 하고, 위·변조하더라도 정족 수 이상의 나머지 블록체인 네트워크 서버가 정상적으로 기능하기 때문에 모니터링 프로세스는 위·변조된 d' 을 검출할 수 있다.

5. 성능 평가

시스템 평가는 블록체인 네트워크에 사용된 4 대의 서버와 분산 저장소로 사용된 7.3.0 버전 Elasticsearch

서버 4 대로 수행되었다. 실험은 수소전기차 공유 업체인 J'CAR[2]의 차량 접속, 탑승, 운행 등의 기록 정보를 가공하여 진행되었다.

<표 1> 시스템 평가에 사용된 서버 사양

운영체제	CPU	메모리
Ubuntu 18.04 Server LTS	Intel™i5 650	4GB

제시한 실험환경에서 한 블록에 속하는 e 의 개수 n 에 변화를 주며 블록 저장 및 블록 검증에 소요되는 시간을 측정한 결과는 표 2 와 같으며 각 실험은 100 번 수행한 값의 평균치이다.

<표 2> 시스템에 대한 저장, 검증, 질의 성능 측정값

n	100	500	1000	2000	5000
블록 저장 (초)	3.829435	3.713685	3.745795	4.063305	4.80284
블록 검증 (초)	1.88676	1.90346	2.23763	2.60806	3.43752
합의 시간 (초)	1.2967	1.3369	1.3652	1.35944	1.3362

실험 결과 한 블록을 저장하는데 약 4 초가 소요되는 것을 확인하였다. 블록 저장에 소요되는 시간의 증가폭은 n 의 증가폭보다 작은 것을 확인할 수 있었다. 각 블록 검증에는 약 2.4 초가 소요되었다. 이 또한 블록 저장의 경우와 마찬가지로 n 의 증가폭에 비하여 소요 시간의 증가폭이 매우 낮은 것을 관찰할 수 있다. 따라서 대용량 데이터 처리에 강점을 보일 것으로 예상한다.

질의에 대한 평가는 질의의 복잡도에 따라 질의 처리 시간이 크게 차이가 날 수 있다. 따라서 본 논문에서는 질의에 대한 시간을 평가하는 대신 질의의 복잡도와 무관하게 오버헤드로 작용하는 합의 시간을 평가했다. 표 2 의 각 n 에 대한 합의 시간은 표 2 의 블록 검증 시에 소요되는 순수한 합의 시간만을 측정한 값이다. 모든 n 에 대하여 합의 시간은 약 1.34 초 내외로 측정되었는데, 이는 블록체인의 합의 과정이 n 의 크기와 관계없이 질의문을 해싱한 값으로 이루어지기 때문이다. 따라서 질의 시에는 질의문을 평가하는 데 소요되는 시간 이외에도 합의를 위한 오버헤드로 약 1.34 초가 소요됨을 확인할 수 있다.

6. 결론

본 논문은 블록체인을 이용하여 분산 저장소의 무결성을 보증하고, 위·변조된 데이터가 존재할 경우 이를 검출하며, 질의 처리부를 두어 데이터 분석기능을 제공하는 실용적 시스템을 제안하였다. 해시트리 전체를 저장하는 대신 ciphertext 의 복호화 결과에 기반한 위·변조 징후 포착 기술 구현을 통하여 공간 효율성을 꾀하였다. 블록체인이 질의 처리부를 관리하도록 하여 단일 중단점 공격을 예방하고, leader election 기반의 마스터 노드 장애 대응을 통하여 안정성을 더했다. 시스템의 성능은 평균적으로 한 블록에 대하여 저장에 4.0 초, 검증에 2.4 초가 소요되었으며 질의의 경우 1.34 초의 합의를 위한 오버헤드가 추가됨을 확인할 수 있었다. 본 논문은 당초 차량 운행 정보를 무결하게 저장하고 이를 안정적으로 질의할 목적으로 개발되었으나, 의료, 사물인터넷 등 여타 다양한 분야에서도 범용적으로 쓰일 수 있을 것으로 기대된다.

[2] http://www.jecar.co.kr

Acknowledgement

본 논문은 2018 년 산업통상자원부의 재원으로 한국산업기술진흥원의 지원 (P0004602)과 2016 년 대한민국 교육부의 재원으로 연구재단의 지원 (NRF-2016R1D1A1B03931324)을 받아 수행된 연구임.

참고 문헌

[1] M. Bertoncello, G. Camplone, P. Gao, H.-W. Kaas, D. Mohr, T. Möller and D. Wee, "Monetizing car data—new service business opportunities to create new customer benefits," *McKinsey & Company,* 2016.

[2] H. C. Kwon, S. J. Lee, J. Y. Choi, B. H. Chung, S. W. Lee and J. C. Nah, "Security Trends for Autonomous Driving Vehicle," *Electronics and Communications Trends,* vol. 33, no. 1, pp. 78-88, 2018.

[3] A. Dorri, M. Steger, S. S. Kanhere and R. Jurdak, "BlockChain: A Distributed Solution to Automotive Security and Privacy," *IEEE Communications Magazine,* vol. 55, pp. 119-125, 12 2017.

[4] A. Dorri, S. S. Kanhere, R. Jurdak and P. Gauravaram, "Blockchain for IoT security and privacy: The case study of a smart home," in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2017.

[5] S. Nakamoto and others, "Bitcoin: A peer-to-peer electronic cash system," 2008.

[6] T. McConaghy, R. Marques, A. Müller, D. De Jonghe, T. McConaghy, G. McMullen, R. Henderson, S. Bellemare and A. Granzotto, "BigchainDB: a scalable blockchain database," *white paper, BigChainDB,* 2016.

[7] E. Gaetani, L. Aniello, R. Baldoni, F. Lombardi, A. Margheri and V. Sassone, "Blockchain-based database to ensure data integrity in cloud computing environments," 2017.

[8] S.-O. Lee, B. Han and H. Han, "Security Assured Vehicle Data Collection Platform by Blockchain: Service Provider's Perspective," in *2019 21st International Conference on Advanced Communication Technology (ICACT)*, 2019.

[9] M. Castro, B. Liskov and others, "Practical Byzantine fault tolerance," in *OSDI*, 1999.

[10] L. Lamport, R. Shostak and M. Pease, "The Byzantine generals problem," *ACM Transactions on Programming Languages and Systems (TOPLAS),* vol. 4, pp. 382-401, 1982.

[11] P. Hunt, M. Konar, F. Junqueira and B. Reed, "ZooKeeper: Wait-free Coordination for Internet-scale Systems.," *USENIX annual technical conference,* vol. 8, no. 9, 2010.