공학석사 학위논문

블록체인 기반의 공정성과 탈중앙화를 지원하는 안전한 데이터 거래 기법

2023년 2월

부 경 대 학 교 대 학 원

정보보호학과

전 미 현

공학석사 학위논문

블록체인 기반의 공정성과 탈중앙화를 지원하는 안전한 데이터 거래 기법

지도교수 신 상 욱

이 논문을 공학석사 학위논문으로 제출함.

2023년 2월

부 경 대 학 교 대 학 원

정보보호학과

전 미 현

전미현의 공학석사 학위논문을 인준함.

2023년 2월 17일

주심 이학박사 이 경 현 (인)

위원 이학박사 신 원 (인)

위원 이학박사 신 상 욱 (인)

목 차

표 목차

그림 목차

Secure data trading scheme with fairness and decentralization based on Blockchain

Mi-Hyeon Jeon

Department of Information Security, The Graduate School,
Pukyong National University

Abstract

Demand for data transactions has increased in various fields with the development of computing technology. Several trading platforms have emerged to cater to this demand; however, most platforms were designed with a centralized model, which has led to performance, security, and privacy concerns. In addition, an increasing number of businesses are using only private platforms for trading to maintain trade secrets. However, these platforms present security and fairness challenges, e.g., denial of their actions by transaction participants and reliance on the trust of trusted third parties (TTPs). To solve these problems, several studies have used blockchain to construct secure data trading platforms. However, in most cases, decentralization is still not satisfied, such as TTP intervention in the initial setting process of the protocol.

This paper proposes a secure and fair data trading protocol using blockchain technology. The proposed protocol eliminates the intervention of TTP by using a randomized arbitrator node selection method for dispute resolution. Furthermore, all-or-nothing transform (AONT) and dual-receiver cryptosystem (DRC) techniques are applied to ensure the requirements for a fair trading platform such as fairness, timeliness and data confidentiality. In addition, by implementing a prototype of the proposed protocol, the existing blockchain technology was compared with fair exchange studies based on it. Compared to other studies, the proposed protocol has yielded results that satisfy certain de-centrality, satisfy confidentiality, and excel in terms of gas consumption.

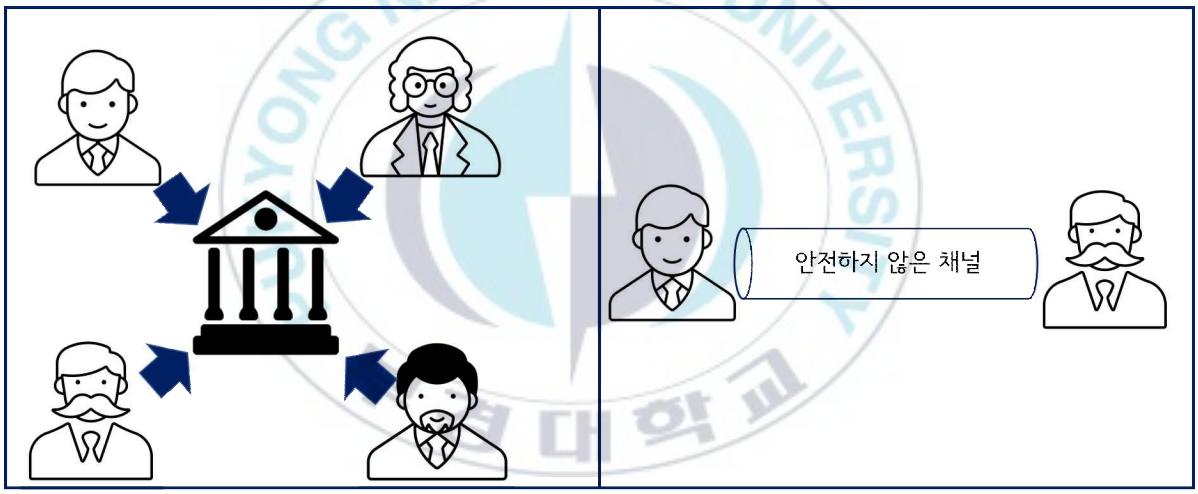
I. 서 론

1. 연구의 필요성

다양한 산업 분야에서 사물 인터넷(IoT, Internet of Things), 블록체인, 클라우드, 인공지능 등 다양한 컴퓨팅 기술 또는 기술들의 결합들이 제안되고 발전되었다. 과학기술정보통신부의 2020년 데이터 산업 현황조사에 따르면 국내 데이터 산업시장은 꾸준히 성장해왔고, 그중에서 특히 데이터 판매 및 제공 서비스업이 48%로 가장 큰 비율을 차지하고 있다는 것을 알 수 있다[1].

현재 많은 산업 분야에서 IoT 장치들을 배치함으로써 생성되는 방대한 양의 데이터는 빅데이터 시대에 점점 더 중요해지고 있으며, 데이터 소유권 관점에도 많은 관심이 쏠리고 있다. 따라서 데이터 소유자(생산자)와 데이터 사용자(소비자) 모두에게 도움이 되는 공정한 데이터 공유/거래 플랫폼은 데이터 중심 사회에서 실질적인 가치를 갖는다. 예를 들어 마이데이터 분야에서 보험 상품을 추천해주는 어플을 만들려는 회사가 고객들의 의료데이터 수집을 원한다고 가정한다면, 데이터 소유자인 고객은 자신의 데이터를 보험사에 제공함으로써 금전적인 이득을 얻을 수 있는 방향으로 발전할 수 있다. 그러나 기존 데이터 공유/거래 솔루션은 [그림 1]에 나타난 것처럼 주로 중앙 집중 플랫폼이나 데이터 소유자와 사용자 간의 비공개 플랫폼을 통해 거래되며, 이는 다음과 같은 문제를 일으킨다. 중앙 집중 플랫폼은 플랫폼 참여자의 계정 관리 및 매칭, 데이터 교환 등 주요 프로

세스를 중앙 집중 기관이 수행하는데, 이 중앙 집중 기관이 단일 실패 지점이 될 수 있다. 또한 백업 혹은 서버 이중화가 되어있지 않다면 공격받았을 때 주요 프로세스를 실행하지 못할 수 있다. 마지막으로 중앙 집중형 모델은 성능, 보안 및 개인 프라이버시 문제 등 본질적인 문제가 있다. 따라서 거래/공유할 데이터의 프라이버시와 거래의 공정성이 신뢰할 수 없는 중앙 집중 기관에 달려 있다는 것은 충분히 우려될 상황이다. 비공개 플랫폼은 개인과 개인 사이에서 거래가 완료되기 위해서는 몇 차례의 협상이 필요하기 때문에 시간, 비용 측면에서 비효율적이고, 개인 간의 거래는 본질적으로 안전하지 않기 때문에, 거래의 보안과 신뢰성을 보장할 수 없다는 문제가 있다.



[그림 1] 기존의 데이터 공유/거래 기법

따라서 공개된 공정한 분산형 거래 플랫폼에 관한 연구가 많은 관심을 받고 있다[2, 3, 4]. 최근 데이터 거래 플랫폼을 탈중앙화 시키기 위해 블록체인을 적용하는 연구가 진행되고 있는데, 블록체인은 제 3자 없이 P2P(Peer-to-Peer) 방식을 기반으로 데이터 블록이 체인 형태를 가진 분산원장 기술로써 다음과 같은 특징을 갖고 있어 기존의 중앙 집중 플랫폼이나 비공개 플랫폼이 갖고 있던 문제점을

해결할 수 있다[5]:

- 탈중앙성: 블록체인은 네트워크 내의 노드들 간에 행해진 거래들이 담겨있는 분산원장을 중앙 집중 서버 혹은 특정 개체에 저장, 관리하지 않고, 참여 노드들이 직접 블록을 생성하고 검증, 합의를 진행한다. 따라서 네트워크 내의 모든 노드들이 동일한 분산원장을 갖고 있다.
- 투명성: 블록체인 네트워크의 채굴자가 블록을 채굴하는 데에 성공하게 되면 채굴자를 제외한 노드들은 해당 블록을 검증하기 시작한다. 검증이 완료된 후 성공적으로 블록체인에 해당 블록이 게시된다. 게시된 블록은 블록체인 참여자 노드들은 모두 확인할 수 있기 때문에 투명성을 보장한다.
- 완결성: 완결성은 하나의 블록이 생성되고, 블록이 체인에 추가되면 취소나 수정이 될 수 없다는 것을 의미하며, 사용자에게 이 거래가 임의로 취소되지 않는다는 확신을 줄 수 있어 완결성은 매우 중요하다. 블록체인에서 완결성은 사용하는 합의 알고리즘이 어떤 것인지에 따라 일부는 확률적인 완결성을 제공한다.

블록체인 기술을 사용한다면 판매자와 구매자 간에 발생하는 거래 정보들이 TTP(Trusted Third Party)의 개입 없이 분산 장부에 저장이 되고, 안전한 거래를 진행할 수 있다. 따라서 본 논문에서는 블록체인 기술을 기반으로 탈중앙화된 공정하고 안전한 데이터 거래 기법을 제안한다.

2. 연구 내용 및 구성

컴퓨팅 패러다임이 독립형(stand-alone) 모델에서 네트워크 지향(network-oriented) 모델로 지속적으로 발전함에 따라 클라우드 컴퓨팅, 엣지 컴퓨팅, 투명 컴퓨팅과 같은 새로운 네트워크 컴퓨팅 기술이 나타나고, 네트워크를 통한 온 디맨드(on-demand) 서비스 제공 방식이 생겨나면서 데이터 혹은 서비스에 대한 거래가 증가하였다[6]. 하지만 이러한 거래 상황에서 판매자 혹은 구매자가 정직하지 않다면, 판매자는 적합하지 않은 데이터나 악의적인 데이터를 제공할 수 있고, 반면에 구매자는 자신의 이익 또는 허위 고발 목적으로 정당한 데이터 획득에 대해 악의적으로 부인할 수 있다. 따라서 거래 시스템에 참여하고 있는 참여자들의 보호를 위해 부인방지 분쟁 메커니즘의 필요성이 대두되고 있다.

부인방지(Non-repudiation)란 참여자가 공유/수신하는 데이터의 유효성과 관련해 자신의 책임에 대한 반박할 수 없는 증거를 제공하는 능력을 말한다[7]. 부인방지 서비스에서 전송자가 수신자에게 어떤 정보를 전달할 때 참여자 중 어느 누구도 이 통신의 전체 또는 부분에 참여한 것을 부인하지 않도록 해야 한다. 따라서 부인방지 프로토콜은 참여자 자신이 통신에 참여했다는 것을 증명하기 위해 전송자는 출처 부인방지 증거(non-repudiation of origin evidences)를, 수신자는 수신 부인방지 증거(non-repudiation of receipt evidences)를 생성해야 한다. 부인 분쟁이 발생하는 경우, 위의 증거들을 기반으로 중재자가 어느 한 개체의 악의적인 행위를 판별해낼 수 있다. 이 부인 분쟁을 해결하기 위해서는 참여자들이 생성한 증거

를 교환해야 하는데, 이는 전자 서명이 등장하면서 공정한 교환을 위한 반박할 수 없는 증거 생성이 가능해졌다. 공정한 교환 프로토콜은 거래에 참여하는 두 당사자 간의 교환 시 당사자 중 어느 쪽도 유리한 상황이 되지 않도록 하는 프로토콜로 원자적(atomic) 특성을 만족한다. 즉, 교환 과정에서 어느 시점이든 프로토콜은 쌍방이 원하는 것을 얻거나, 아니면 어느 쪽도 아무것도 얻지 못한다는 것을 보장한다는 것이다.

공정한 부인방지 프로토콜은 TTP의 개입 여부에 따라 분류될 수 있으며, TTP의 지원을 받는 프로토콜은 다시 TTP의 개입 정도에 따라 인라인(Inline), 온라인(Online), 오프라인(Off-line) 프로토콜로 분류할 수 있다.

● TTP의 지원을 받지 않는 프로토콜

- 확률적 접근법: 이 접근법은 거래의 참여자 사이의 공정성을 확률론적으로 보장하는 대신 TTP의 개입을 피하는 프로토콜로 참여자 사이에 계속적인 통신으로 한 개체가 다른 개체보다 더 많은 특권을 가지지 않도록 하는 것이다. 하지만 이 방법으론 진정한 공정성이 보장되지 않으며, 두 명의 참여자가 동일한 계산능력을 가져야만 공정성이 보장된다는 비실현적인 가정에 기반을 둔다. 또한 공정성을 실현하기 위한 계속적인 통신은 많은 시간과 통신 오버헤드를 발생시킨다는 단점이 있다.

● TTP의 지원을 받는 프로토콜

– 인라인 TTP 프로토콜: TTP는 부인방지 서버, 거래의 중재자로 작동하며, 수신자와 송신자는 모두 이 서버를 거쳐 메시지, 데이터를 공유한다. 또한 TTP가 거래의 중재자로 참여하면서 부인방지 증거를 수집해 수신자와 송신자에게 전달함으로써 부인 분쟁을 해결한다. 이 프로토콜은 TTP가 관리하는 대규모 데이터베이스가 공격받으면 피해가 매우 크며, 병목현상이 발생할 수 있는 단점이 있다.

– 온라인 TTP 프로토콜: 인라인 TTP와 다르게 수신자와 송신자 사이의 각 전송에 중재자로 작동하지 않지만, 프로토콜에 정해진 각 세션 동안에만 개입한다[8]. 이는 TTP의 부하를 줄이기 위해 제안된 방식이지만 여전히 TTP의 과도한 신뢰성 문제, 병목현상, 단일 실패 지점 문제가 존재한다.

– 오프라인 TTP 프로토콜: 이 프로토콜은 참여 개체가 정직하지 않은 행위를 하거나 네트워크에 문제가 발생했을 때만 TTP가 개입하는 프로토콜이다[9]. 정직하지 않은 개체가 존재한다면 반대쪽 참여자가 공정한 방법으로 프로토콜을 종료하기 위해 도움을 주도록 TTP를 호출한다. 이 프로토콜은 거래 과정의 대부분 시간에서 문제가 발생하지 않는다고 가정하기 때문에 낙관적(optimistic) 프로토콜이라고도 불린다. 오프라인 TTP 프로토콜은 인라인, 온라인 TTP 프로토콜, 확률적 접근법이 가진 문제들 대부분을 해결하지만, TTP 신뢰도 문제는 여전히 해결하지 못한다는 단점이 있다.

II. 연구 배경

본 장에서는 본 논문의 관련 연구인 리더 선정 기법과 제안 프로토콜에 적용된 암호학적 프리미티브에 대해 설명하고, 기존의 부인방지 관련 연구에 대해 살펴본다.

1. Algorand의 리더 선정 기법

Chen 등이 랜덤 비콘 프로토콜을 활용해 블록체인 네트워크 내의 어떤 노드가 합의 노드가 될지를 예측 불가능하게 하는 리더 선택 기법을 제안하였다[10]. 제안 프로토콜에서 랜덤하게 선정된 합의 노드가 부인 분쟁 발생 시 중재자의 역할을 하게 된다. 리더 선정 기법은 랜덤 비콘 값을 입력 값으로 하여 다음 거래를 위한 랜덤 비콘 값을 유도한다. $r-1$번째 DO(Data Owner, 데이터 소유자)의 랜덤 비콘 값 Q_{DO}^{r-1}이 주어졌을 때, 다음 라운드의 랜덤 비콘 값 Q_{DO}^{r}을 유도하는 방법은 2가지가 존재한다. 첫 번째 방법은 미리 정의된 시간 간격 동안 선정된 리더 노드가 자신의 중재자 자격 증명서와 이전 라운드의 랜덤 비콘 값의 서명 $SIG_{l_{DO}^{r}}(Q_{DO}^{r-1})$을 함께 공개하는 것이다. 그 후 $r-1$ 값과 함께 해시값을 구하는 것으로 다음 식과 같이 다음 라운드의 랜덤 비콘 값을 도출할 수 있다.

$$Q_{DO}^{r} = H(SIG_{l_{DO}^{r}}(Q_{DO}^{r-1}), r-1)$$

위의 식을 통해 다음 라운드의 랜덤 비콘 값을 유도할 수 없다면 아래의 식으로 랜덤 비콘 값을 도출한다.

$$Q_{DO}^{r} = H(Q_{DO}^{r-1}, r-1)$$

2. 암호학적 프리미티브

가. Dual-Receiver Cryptosystem(DRC)

DRC 기술은 서로 다른 이중 수신기에서 추출한 키를 사용해 암호화와 서명을 진행하는 기술로써 선택 암호문 공격에 대해 안전한 암호화 체계와 적응형 선택 메시지 공격에 대해 안전한 서명 기법으로 구성된 결합 공개키 암호시스템이다[11]. 대부분의 공개키 암호시스템은 하나의 개인키로 암호화와 서명을 진행하게 되는데, 이는 문제 발생 시에 제 3자가 데이터를 복원할 수 없게 된다. 이상적인 암호시스템은 공적인 사용을 위한 복구 가능한 암호문과 사적인 사용을 위한 복구 불가능한 암호문을 만들 수 있어야 하므로 DRC 알고리즘은 이를 고려해 총 6개의 알고리즘으로 구성되어있다:

$DRC_{comb} = (\kappa, \varepsilon, D, R, S, V)$.

- 키 생성 알고리즘 κ: 보안 파라미터 k를 입력으로 사용해 2개의 키 쌍을 생성한다. 각각의 키는 암호화와 서명을 위한 키이다.
- 암호화 알고리즘 ε: 각각 다른 이중 수신기로부터 추출한 공개 암호화키와 메시지 $m \in M$을 입력 값으로 사용해 암호문 $c \in C$을 생성하는 무작위 알고리즘이다.
- 복호화 알고리즘 D: 복호화를 실시하는 수신기의 개인키와 암호화를 진행한 공개키, 암호문 $c \in C$을 입력으로 사용해 $m \in M$ 또는 거부 기호를 생성하는 결정론적 알고리즘이다.

- 복구 알고리즘 R: 암호화를 실시하는 수신기의 개인키와 다른 수신기의 공개키, 암호문 $c \in C$을 입력으로 사용해 $m \in M$ 또는 거부 기호를 생성하는 결정론적 알고리즘이다.
- 서명 알고리즘 S: 개인 서명키와 메시지 $m \in M$을 입력으로 사용해 서명 $\sigma \in \{0,1\}^*$을 생성하는 무작위 알고리즘이다.
- 검증 알고리즘 V: 공개 검증키와 메시지, 서명 쌍 (m, σ)을 입력으로 하여, 수락 혹은 거절 출력 값을 생성하는 무작위 알고리즘이다.

DRC 알고리즘은 키가 하나의 개체에서 생성되지 않고, 서로 다른 이중 수신기에서 생성되기 때문에 블록체인을 활용한 탈중앙형 서비스 환경에 적합한 특징을 지닌다. 따라서 데이터 소유자가 데이터 요구자에게 거래에 필요한 중요 정보들을 전달할 때 데이터 요구자의 공개키와 중재자 노드의 공개키를 이용해 암호화를 진행한다. 데이터 요구자는 중요 정보를 복호화하기 위해 자신의 개인키와 중재자 노드의 공개키를 사용해 중요 정보를 획득할 수 있다. 또한 부인 분쟁 중재 요청받은 중재자 노드 역시 중요 정보 획득을 위해 중재자 노드의 개인키, 데이터 요구자의 공개키를 복구 알고리즘의 입력 값으로 사용해 데이터를 복구할 수 있다.

나. All-Or-Nothing Transform(AONT)

AONT는 메시지를 암호화하기 전에 적절한 사전 처리를 통해 무차별 검색 공격자가 암호문 한 블록만 복호화하여 일반 텍스트 블록을 얻을 수 있는 것을 방지하는 기술이다[12]. 일반적으로 단일 암호 텍스트 블록을 복호화 가능하다면 악의적인 공격자가 가진 후보키를 테스트하기에 충분하므로 키 검색 문제를 비교적 쉽게 만들 수 있

다. 이러한 특징을 분리성이라고 하는데, AONT는 비분리 특성을 채택하여 무차별 검색 공격을 어렵게 만든다. 따라서 AONT는 모든 암호 텍스트 블록을 복호화하지 않는 이상 어떤 단 하나의 일반 텍스트 블록도 얻어지지 않는다는 속성을 보장한다.

AONT의 간단한 동작은 아래에 서술된다:

- 메시지 시퀀스 $m_1, m_2, \ldots, m_s$를 AONT를 통해 의사 메시지 시퀀스 $m_1', m_2', \ldots, m_s'$로 변환한다.
- 주어진 암호화키 K로 일반 암호화 모드로 의사 메시지 시퀀스를 암호화하여 암호 텍스트 시퀀스 $c_1, c_2, \ldots, c_t$를 획득한다.

AONT를 수행하기 위해서는 아래의 속성으로 정의되어야 한다.

(정의) 메시지 시퀀스 $m_1, m_2, \ldots, m_s$를 의사 메시지 시퀀스 $m_1', m_2', \ldots, m_s'$로 매핑하는 변환 f는 아래의 속성을 만족해야 AONT라고 할 수 있다.

- 변환 f는 가역적이다: 의사 메시지 시퀀스가 주어진다면 원본 메시지 시퀀스를 얻을 수 있다.
- 변환 f와 그의 역변환은 모두 효율적으로 계산할 수 있다.
- 의사 메시지 블록 중 하나를 알 수 없는 경우 원본 메시지 블록의 함수를 계산하는 것은 계산적으로 불가능하다.

3. 관련 연구

현재 블록체인 기술을 기반으로 하는 부인방지 연구와 공정한 데이터 공유를 지원하는 연구들이 많이 제안되었다. Ferrer 등은 비트코인 플랫폼을 활용한 블록체인 기술로 거래에 참여하는 두 참여자 중

어느 쪽도 유리한 상황 혹은 불리한 상황에 처하지 않는 프로토콜을 제안하였다[13]. Huang 등의 연구에서는 블록체인 기반 3자 간 공정한 계약 서명이 가능하게 하는 계약 서명 프로토콜을 제안하였다[14]. [14]의 연구에서는 참가자 한 명이 계약을 제시하면 나머지 참가자 두 명의 서명을 받아야만 3자 간의 계약이 유효하게 만들어 계약의 공정성을 제공하였다. Li 등은 대화형 서비스에서 발생하는 부인을 방지하기 위해 디지털 인증서 형태로 통신 증거를 생성하고, 통신 과정에 참여하는 참여자를 블록체인 네트워크 노드의 역할도 겸하게 하여 통신 과정에서 생성한 정보를 사용자 노드가 직접 수집해 블록체인 네트워크에 게시해 부인방지 증거로 사용하게 하였다[15]. Xu 등은 IIoT(Industrial IoT) 시나리오에서 네트워크 컴퓨팅 서비스를 제공하기 위해 블록체인 기술을 사용하였다[16]. IIoT 클라이언트가 블록체인에 서비스를 요청하면 서비스 제공자는 프로그램을 크기가 다르게 두 부분으로 나누어 거래가 정상적으로 완료되었을 때만 프로그램 전체를 획득할 수 있게 설계해 클라이언트의 부인을 방지하였다. Ahmad 등의 연구에서 판매자는 판매할 데이터에 대해 유연한 접근통제 정책을 수립할 수 있고, 구매자는 구입한 데이터에 대해 고급 zk-SNARKs 프로토콜을 사용해 정보를 공개하지 않고도 암호화된 데이터의 정확성을 검증할 수 있는 프로토콜을 제안해 공정한 데이터 교환을 실현하게 했다[17]. Li 등의 연구에서 블록체인, 스마트 컨트랙트, 클라우드 서비스를 활용한 공개된 공정 데이터 거래를 위한 분산 거래 솔루션을 제안하였다[18]. 판매자는 판매할 데이터를 스토리지에 암호화한 후 업로드하고, 구매를 원하는 구매자에게 데이터를 복호화하는 데 사용되는 키를 off-chain 방법으로 전달함으로써 안전한 저장 및 효율적인 교환을 실현할 수 있다.

III. 블록체인 기반의 공정하고 안전한 데이터 거래 프로토콜

본 장에서는 탈중앙화된 환경에서 공정하고 안전한 데이터 거래 기법을 제안한다. 먼저, 제안 프로토콜의 가정과 이를 고려한 위협모델에 관해 서술한다. 그리고 제안 프로토콜의 구성요소와 프로토콜 설계 시 기반이 되는 암호학적 프리미티브에 대해 자세히 설명한 후 구체적인 프로토콜 실행과정과 스마트 컨트랙트를 통해 제안 프로토콜을 정의한다.

1. 가정 및 위협모델

제안 프로토콜은 공개적인 공정하고 안전한 프로토콜을 위해 퍼블릭 블록체인 네트워크를 가정한다. 프로토콜에 사용되는 블록체인(스마트컨트랙트 포함)은 기반이 되는 블록체인 플랫폼의 안전성을 상속받는다. 즉, 기반이 되는 블록체인의 무결성, 변조 불능성 등의 기본 안전성을 제공받는다.

제안 프로토콜의 중재자 노드의 역할을 하는 노드는 완전 신뢰 개체가 아니다. 따라서 중재자를 선정하는 관정을 랜덤하게 하여 블록체인 노드가 특정 거래에 의도적으로 참여하여 부정한 행위를 할 수 없도록 제안한다. 또한 중재자 노드 선정 기법을 사용하기 위해 중재자 노드가 되고자 하는 노드의 수가 충분하다고 가정한다.

공격자는 프로토콜 실행과정 이전에만 어떤 개체를 붕괴시킬 수 있는 static corruption을 가정한다[18]. 추가적으로 공격자가 계산량적으로 제한된 능력을 가진다고 가정한다. 즉, Probabilistic Polynomial-Time (P.P.T) 알고리즘으로 제한된다. 또한 제안 프로토콜은 통신을 제어하는 공격자의 능력을 기술하기 위해서 인증된 P2P(Point-to-Point) 채널의 동기화된 네트워크 모델을 가정한다 (synchronous authenticated channel). 즉, 정직한 당사자 간에 전달되는 어떤 메시지에 대해 공격자는 사전에 알려진 Δ까지 지연시키지만, 삭제, 경로 변경, 수정은 할 수 없다고 가정한다. 일반성을 잃지 않고, 시스템에 글로벌 클럭을 고려한다[19].

본 논문의 위협모델은 다음과 같이 가정한다. 거래 참여 개체인 DO(Data Owner)와 DR(Data Requester)은 서로 신뢰하지 않는다. 즉, DO와 DR 모두 상대방을 속이고 자신의 이득을 취하려는 행위를 할 수 있다[20]. DO의 경우, DR로부터 데이터 구매에 대한 금액을 받고, 데이터를 전달하지 않거나 거짓/잘못된 데이터를 전달하려고 할 수 있다. DR은 데이터에 대한 대금을 지불하지 않고 DO로부터 데이터를 획득하려고 시도할 수 있다. 블록체인 중재자 노드로 선정된 노드는 정직하지 않거나 호기심이 많을 수 있다. 분쟁 해결 시에 부인 증거를 스마트 컨트랙트(SC)에 보내지 않거나 올바르지 않은 정보를 보내려고 시도할 수 있다.

또한 내부 결함에 의한 위협이 있을 수 있다. 이러한 위협은 플랫폼의 사용자 혹은 관리자 등 사람이 예측할 수 없고, 진정한 공정성에 큰 영향을 끼치게 된다. 내부 결합에 의한 위협으로는 플랫폼의 네트워크 지연, 가용 자원 고갈 등의 내부 오류로 인해 발생하는 트랜잭션의 오류가 부인방지 프로토콜의 증거 생성 및 검증에 영향을

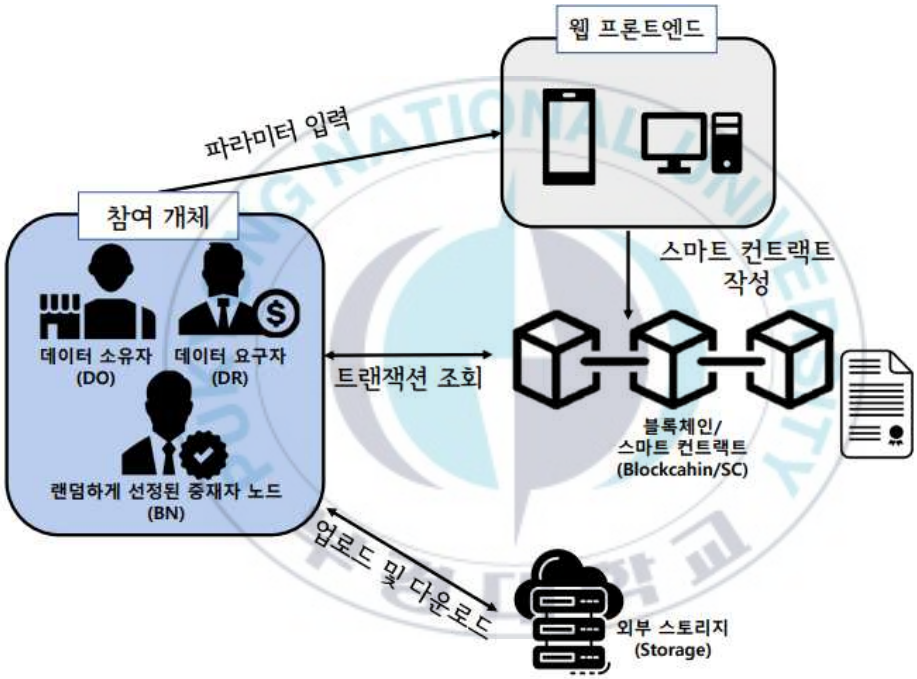
미친다.

따라서 본 논문은 다음과 같은 요구 사항을 만족시키는 것을 목표로 한다.

- 트랜잭션의 모든 참여 개체가 정직하다는 조건으로, 동기 인증된 네트워크와 독립 실행형 모델에서 완전성을 달성한다.
- 동기 인증된 모델과 독립 실행형 설정에서 기본 암호화 기본 요소가 안전한 조건으로 DO 또는 DR 중 하나가 비적응 P.P.T 공격자에 의해 손상되더라도 공정성 요구 사항을 충족해야 한다.
- 공격자와 중재자 노드 사이의 담합 공격으로부터 안전해야 한다.
- 동기 인증된 네트워크 및 독립 실행형 모델에서, 비적응적 P.P.T 공격자에 대한 기밀성을 충족해야 한다.
- DO 혹은 DR 중 하나라도 정직하다면 동기 인증된 네트워크 및 독립 실행형 모델에서 적시성(Timeliness)을 충족해야 한다.

2. 제안 프로토콜

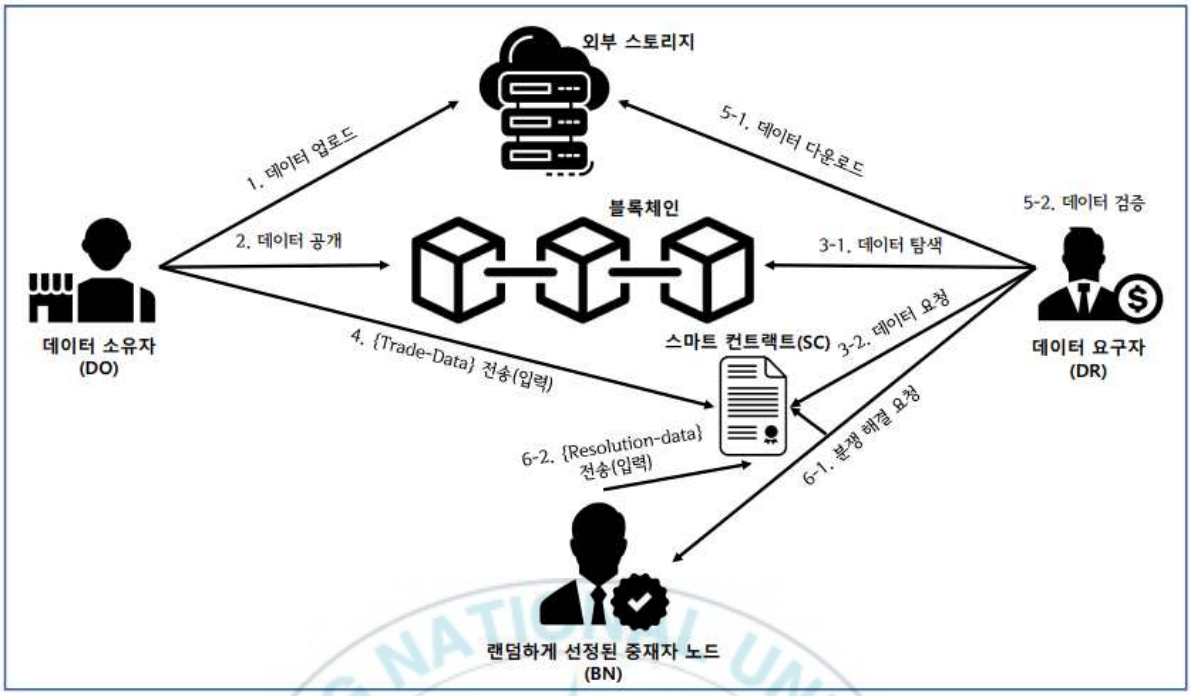
[그림 2]는 제안 프로토콜의 개요를 나타낸다. 웹 프론트엔드는 참여 개체들로부터 파라미터들을 입력받고, 이를 기반으로 스마트 컨트랙트를 작성한다. 스마트 컨트랙트를 통해 블록체인에 게시된 트랜잭션들을 참여 개체들이 조회할 수 있다.



[그림 5] 제안 프로토콜 개요

가. 구성요소

제안 프로토콜은 총 5개의 구성요소로 구성되어있다. [그림 3]을 통해 구성요소 간의 상호작용이 어떻게 이루어지는지 나타나고, 각 구성요소의 역할이 다음에 서술된다. [표 1]을 통해 제안 프로토콜에



[그림 6] 제안 프로토콜의 구성요소

서 사용된 용어를 설명하였다.

데이터 소유자(DO)는 자신이 소유하고 있는 데이터를 원하는 개체와 유연한 접근 제어 정책을 통해 거래하기를 원하므로 블록체인과 P2P 네트워크 참여자로 가정한다. 데이터 소유자는 자신의 ID와 각각의 데이터 거래에 대해 일회용 키 쌍(PK_{DO}, SK_{DO}) 을 가지고 있다고 가정하며 성공적인 데이터 거래에 대한 수수료 Fee_{BN} 지불을 약속함으로써 제3자(viz. 중재자 노드)에게 공정한 교환의 중재를 원한다.

데이터 요구자(DR)는 자신이 원하는 데이터를 소유한 DO로부터 데이터를 구매하기를 원한다. 이를 위해 블록체인과 P2P 네트워크에 참여한다. 데이터 요구자 역시 자신의 ID와 각각의 데이터 거래에 대해 일회용 키 쌍(PK_{DR}, SK_{DR})을 가지고 있다고 가정하고 데이터 구매에 대해 데이터 소유자에게 $Price_M$을 지불한다.

[표 1] 제안 프로토콜의 용어 및 약어

용어 및 약어	상세
ID_x	참여자 x의 ID(주소)
M	데이터(평문)
PK_x	참여자 x의 일회용 공개키
SK_x	참여자 x의 일회용 개인키
$Stub$	M의 $AONT$변환 후 짧은 부분
$Package$	M의 $AONT$변환 후 긴 부분
σ_x	x의 서명
I_{Stub}	DRC 사용 후 암호화된 값
Fee_{BN}	BN에게 지불되는 거래 수수료
$Price_M$	M에 대한 가격
$H(x)$	x의 해시 값

중재자 노드(BN)는 블록체인 네트워크에서 합의에 참여하는 개체로, 중재자 노드의 선정은 사용된 블록체인 인프라에 의존한다. 퍼블릭 블록체인 모델인 경우, 네트워크 내의 모든 참여 개체가 BN으로 선정될 수 있고, 프라이빗 블록체인 모델인 경우에는 선택된 몇몇 개체가 BN으로 참여한다. BN은 탈중앙화된 공정한 데이터 거래 프로토콜을 지원하기 위해 데이터 거래에 대한 중재 수수료 Fee_{BN}를 받고 분쟁 해결을 수행한다. 만일 BN이 부정하게 행동한다면, 중재자 노드로 선정되기 위한 보증금 $Deposit_{BN}$이 몰수된다. 중재자 노드 역시 자신의 ID와 각 데이터 거래에 대해 일회용 키 쌍을 사용한다.

블록체인/스마트 컨트랙트는 안전하고 신뢰성 있는, 그리고 스마트 컨트랙트를 지원하는 블록체인 인프라가 사용된다고 가정한다. 또한 참여 개체들 간의 통신을 용이하게 하기 위해 P2P 네트워크가 사용된다. 스마트 컨트랙트는 참여 개체(공격자를 포함한다)에게 모든 내부 상태를 노출하는 'stateful ideal functionality' 이다[21]. SC는 암호화폐 원장 상에서 코인으로 트랜잭트하기 위해 어떤 불변성 조건을 미리 명시하여, 실제 생활에서 컨트랙트를 투명하게 모방할 수 있다. 실제 컨트랙트는 이더리움과 같은 블록체인을 통해 인스턴스화될 수 있다.

스토리지는 데이터 소유자가 데이터를 저장하는 공간으로 클라우드 스토리지(Cloud storage) 또는 IPFS와 같은 분산 스토리지가 사용될 수 있다. 스토리지는 신뢰성 있게 데이터를 관리하며, 가용성을 보장해야 한다. 스토리지 역시 완전 신뢰 개체가 아니므로 데이터 소유자는 데이터를 스토리지에 아웃소싱하기 전에 암호화를 시켜 기밀성을 보장해야 한다.

나. 암호학적 프리미티브

제안 프로토콜은 2장에 소개된 DRC와 AONT 기술을 사용하고 아래에 서술되는 암호학적 프리미티브를 사용한다.

(a) 암호학적으로 안전한 해시 함수 $H: \{0,1\}^{*} \rightarrow \{0,1\}^{\lambda}$. 여기서 λ는 보안 파라미터이다.

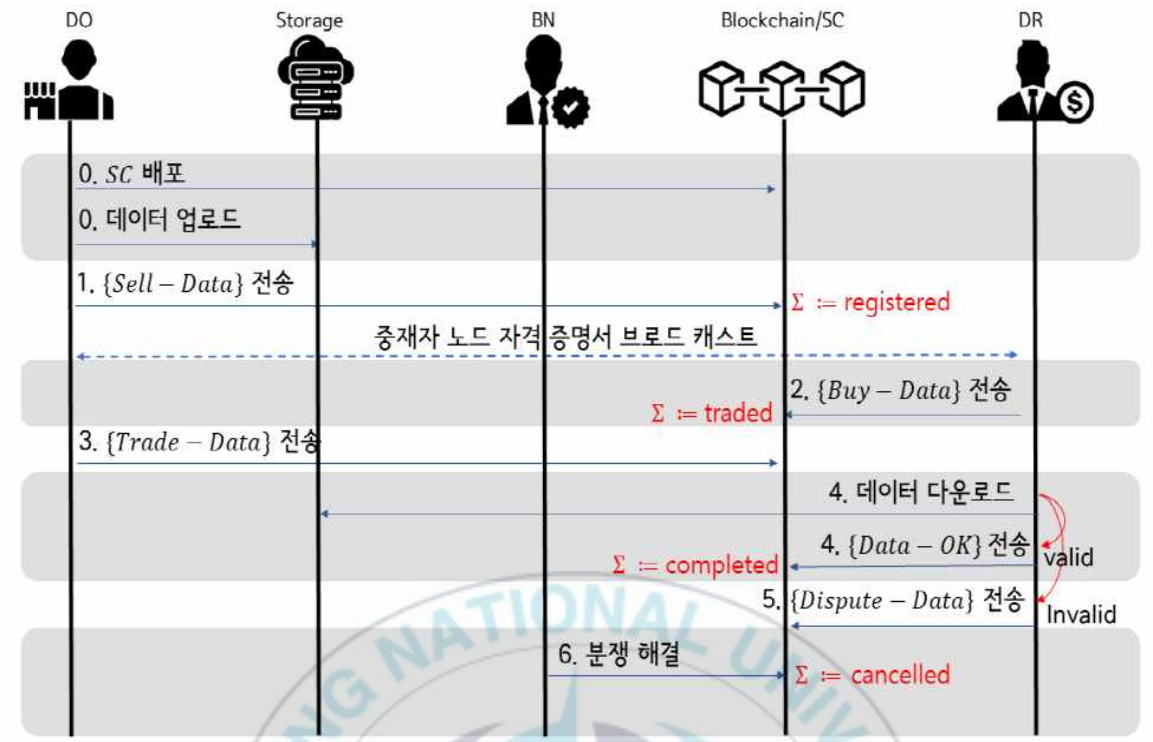
(b) $(SymkGen, SymEnc, SymDec)$로 구성된 의미론적으로 안전한(semantically secure) 대칭 암호 알고리즘

(c) $(AsymKGen, AsymEnc, AsymDec)$로 구성된 선택 암호문 공격에 대해 의미론적으로 안전한 비대칭 암호 알고리즘

(d) 다항식 시간 알고리즘$(SignKGen, Sign, Verify)$로 구성된 선택 메시지 공격에서 존재론적으로 위조 불가능한(existential unforgeability under chosen message attack, EU-CMA) 안전한 전자 서명 기법

3. 구체적인 프로토콜 실행과정

전체적인 서비스 프로토콜은 [그림 4]에 나타나 있다. 제안 프로토콜은 2장에 서술된 암호학 기법을 적용하여 구성되며, '사전 준비, '데이터 거래 공개(Publish)', '데이터 요청(Buy-request)', '데이터 거래(Trading)', '데이터 다운로드 및 검증(Download & Validate)', '분쟁 해결(Dispute resolution request)' 단계로 진행된다. 제안 기법은 데이터 소유자와 데이터 요구자 중 한 개체가



[그림 7] 제안 프로토콜의 순서도

비적응적 P.P.T. 공격자에 의해 손상되더라도 각 당사자에 대해 공정성을 보장한다. [그림 5] ~ [그림 9]는 단계별로 SC가 실시하는 내용을 의사 코드로 나타낸 것이다. 제안 프로토콜에 기술된 SC는 공정한 데이터 거래를 지원하기 위해 블록체인에 액세스를 가지는 stateful ideal functionality이며, SC의 관례적인 의사 코드를 따라서 기술되었다. 튜링 완전 스마트 컨트랙트가 미리 명시된 기능성을 투명하게 다루는 stateful 프로그램으로 볼 수 있으며, 또한 SC가 자체 내부 저장 상태에 따라 조건부 지불을 충실하게 처리하는 암호화폐 블록체인에 액세스할 수 있기 때문에 이러한 표기는 현실의 SC를 반영한다. 기존 방법들과 달리 제안 기법의 SC에서는 복잡하고 무거운 연산을 실행하지 않고, 비교, 할당, 저장, 가감산 연산과 같은 비교적 가스 소모량이 많지 않은 연산들로만 구성되어있다. 복잡한 암호학적 연산들은 참여 개체들이 로컬로 수행한다.

가. 사전 준비 단계

데이터 소유자(DO)는 가장 먼저 스마트 컨트랙트를 배포한다. 그 후 거래 혹은 공유할 데이터 M을 AONT 처리를 통해 $(Stub \| Package)$쌍으로 변환시킨다. 여기서 비교적 짧은 길이의 Stub 부분을 일회용 데이터 키 TK를 사용해 AES와 같은 대칭키 암호 알고리즘을 적용하여 $eStub$으로 암호화한다: $eStub = SymEnc(TK, (Stub \| H(SK_{DO}))$, 여기서 $H(SK_{DO})$는 AONT 변환 시 사용된다. DO는 $(Package, \sigma_{pkg})$를 외부 스토리지(Cloud storage 또는 IPFS)에 저장하며, 저장된 주소를 $Addr_{DO-M}$이라고 한다. σ_{pkg}는 Package에 대한 서명 값이다: $\sigma_{pkg} = Sign(Package, SK_{DO})$. DO가 실시한 암호화 및 서명 과정은 데이터 M에 대한 거래의 일회용 키 쌍(PK_{DO}, SK_{DO})을 사용한다.

나. 데이터 거래 공개(Publish) 단계

DO는 데이터 M에 대한 메타데이터를 포함한 파라미터들을 컨트랙트 SC에 입력(전송)한다:

$$\{Sell,\ ID_{DO}, H(M), H(Stub \| Package), Keywords_M, Terms_M\}$$

ID_DO와 같은 단계별 공통 파라미터들은 [표 1]에 자세히 서술되어있다. Publish 단계에서 $Keywords_M, Terms_M$은 DR이 데이터 검색 시 사용하는 것으로 각각 키워드와 조건의 역할을 한다. (r, Q)는 데이터 거래의 중재자 노드 선정 시 필요한 파라미터로 DO가 선정한다. 이 단계에서 정의되는 두 가지의 타임스탬프 중 $T1$은 중재자 노

드가 되고 싶어 하는 노드들이 중재자 노드로 신청하는 deadline을 뜻한다. $T2$는 자신이 판매/공유하고자 하는 데이터 M에 대한 거래 요청 deadline을 뜻한다.

Algorithm 1: Publish

Input: $\{Sell-Data\}$
SC receives $\{Sell-Data\}$ from DO;
if *DO's account balance* $\geq (Fee_{BN} + Deposit_{DO})$ *and T1, T2 are valid* **then**
  store $\{Sell-Data\}$;
  assert $\Sigma \equiv$ 'registered';
  send $\{Sell-Data\}$ to all nodes;
else
  Terminate;
end

[그림 8] 데이터 공개 단계의 스마트 컨트랙트

SC는 DO로부터 $\{Sell-Data\}$를 수신하고 [그림 5]에서 표현되었듯이 DO의 계정 잔액이 BN에게 지불하는 수수료와 보증금의 합 이상인지 확인한 후에 타임스탬프 $T1$과 $T2$를 확인한다. 그 후 상태 변수 Σ를 'registered' 로 설정하고 SC에 저장할 파라미터들을 저장한다. 모든 과정이 끝나면 네트워크 내의 노드들에게 $\{Sell-Data\}$를 브로드캐스팅한다.

중재자 노드 A가 SC로부터 $\{Sell-Data\}$를 수신하면 중재자 자격 증명서 $Sign_A(r,Q)$를 $T1$ 이내에 브로드캐스트한다. DO는 $T1$이내에 브로드캐스트된 중재자 자격 증명서들 중에서 $.H(Sign_A)$가 가장 작은 노드를 해당 데이터 거래 중재자로 선택한다.

다. 데이터 구매 요청(Buy-request) 단계

DR은 블록체인에 게시된 메타데이터 검색을 통해 원하는 데이터를

조회한 후, 데이터 거래를 위한 파라미터를 SC에게 입력(전송)한다. DR 역시 데이터 M의 거래에 사용되는 일회용 키 쌍 (PK_{DR}, SK_{DR})을 사용한다.

$$\{Buy,\ ID_{DR},\ H(M),\ Price_M,\ Deposit_{DR},\ PK_{DR},\ T3\ \}$$

여기서 $T3$는 거래가 끝날 때까지의 deadline을 뜻한다.

Algorithm 2: Buy-request

Input: $\{Buy - Data\}$
SC receives $\{Buy - Data\}$ from DR;
if *DR's account balance* $\geq (Price_M + Deposit_{DR})$ *and* Σ *is 'registered'* **then**
  store $ID_{DR}, Deposit_{DR}, PK_{DR}, T3$ on SC;
  assert $\Sigma \equiv$ 'req-buy';
  send $\{Buy - Data\}$ to all nodes;
else
  Terminate;
end

[그림 9] 데이터 구매 요청 단계의 스마트 컨트랙트

SC는 DR로부터 구매 요청에 관한 정보를 수신하면 다음 [그림 6]과 같이 동작한다. 상태 변수 Σ가 'registered' 인지 확인하고, 거래 요청이 $T2$이내인지, DR의 계정 잔액이 $(Price_M + Deposit_{DR})$ 이상인지 확인한다. 검증이 끝나면 Σ를 'req-buy'로 설정하고 SC에 파라미터들을 저장한다. 그 후 DR로부터 받은 $\{Buy - Data\}$를 모든 노드들에게 전송한다.

라. 데이터 거래(Trading)단계

DO가 SC로부터 $\{Buy - Data\}$를 수신하면, Σ가 'req-buy' 이고, 타임스탬프 $T2$ 이내인지 확인한다. DR이 다운받은 데이터를 복호화할 수 있게 필요한 파라미터들을 준비한다. 첫 번째로 임시 데이

터 키 TK를 DR의 공개키로 암호화한다: $eTK = AsymEnc(PK_{DR}, TK)$. 그 후 DR이 데이터의 유효성을 검증할 수 있도록 DO의 개인키로 서명 σ_{Stub}을 계산한다: $\sigma_{Stub} \Leftarrow Sign((Stub, TK), SK_{DO})$. 마지막으로 DR과 A의 공개키를 기반으로 DRC 암호화 알고리즘을 사용해 I_{Stub}을 계산한다: $I_{Stub} = DRC_{Enc}((eTK, Addr_{DO-M}), PK_{DR}, PK_A)$. DO가 로컬로 실행한 암호화 과정이 끝나면 {Trade−Data}를 전송(입력)한다: $\{Trade, ID_{DO}, ID_{DR}, ID_A, H(M), eStub, H(eStub), \sigma_{Stub}, I_{Stub}, T4\}$.

여기서 $T4$는 DR이 중재자 노드 A에게 부정한 데이터를 획득했을

Algorithm 3: Trade

Input: $\{Trade - Data\}$
SC receives $\{Trade - Data\}$ from DO;
if Σ *is 'req-buy' and current time is within T3* **then**
  verify other parameters;
  assert $\Sigma \equiv$ 'traded';
  store $ID_A, eStub, H(eStub), \sigma_{Stub}, I_{Stub}, T4$;
  send $\{Trade - Data\}$ to all nodes;
else
  assert $\Sigma \equiv$ 'cancelled';
end

[그림 10] 데이터 거래 단계의 스마트 컨트랙트

때 분쟁 해결을 요청할 수 있는 deadline을 의미한다.

SC는 DO로부터 $\{Trade - Data\}$를 수신한 후 [그림 7]과 같이 동작한다. Σ가 'req−buy' 인지 확인하고, 거래 요청이 $T3$이내 인지 확인한다. 만약 $T3$가 만료된 후에 DO로부터 $\{Trade - Data\}$를 수신하였다면 거래를 즉시 취소하고, Σ를 'cancelled' 로 변경한다. 거래 요청이 $T3$이내면 $T4$를 확인하고 Σ를 'traded' 로 설정하고

모든 노드들에게 $\{Trade-Data\}$를 전송한다.

마. 데이터 다운로드 및 구매 확인(Download & Validate) 단계

SC로부터 $\{Trade-Data\}$를 수신한 DR은 현재 타임스탬프가 $T3$ 이내인지 그리고 Σ가 'traded' 인지 확인한다. 검증이 끝나고 DR은 I_{Stub}을 복원하기 위해 자신의 개인키와 A의 공개키를 사용한다: $(eTK, Addr_{DO-M}) = DRC_{Dec}(I_{Stub}, SK_{DR}, PK_A)$. DR은 획득한 외부 스토리지 주소 $Addr_{DO-M}$을 통해 $(Package', \sigma'_{pkg})$를 다운로드 후, 서명 σ_{pkg}를 검증한다. 다음으로 암호화된 일회용 키 eTK를 TK로 복원하고, 이 키를 사용해 $eStub$을 복호화하여 $(Stub' \| K')$을 복원한 후, $\{Trade-Data\}$로부터 서명 σ_{Stub}을 검증한다. 이 과정을 통해 TK'와 $Stub'$의 유효성을 검증할 수 있다. 이 검증 과정이 끝나면 $H(Stub' \| Package)$을 검증하고, 마지막으로 $(Stub' \| Package)$쌍으로부터 AONT 역변환을 통해 데이터 M'을 복원한 후, $H(M')$을 검증한다. 복원된 데이터 M의 유효성 검증이 성공하면, DR은 SC에게 $\{Data-OK, H(M)\}$을 입력(전송)한다.

SC는 이후에 [그림 8]과 같이 동작한다. Σ가 'traded' 인지 확인하고, $\{Data-OK, H(M)\}$를 수신하거나 $T4$가 만료된 경우 노드 DR의 계정 잔액을 $Price_M$만큼 차감하고, DO에게 지불한다. 모든 행위가 끝나면 상태 Σ를 'completed'로 설정하고, 모든 노드에게 $\{Completed-Data\}$를 전송한다:

$$\{Completed, ID_{DO}, ID_{DR}, ID_A, H(M)\}$$

Algorithm 4: Download & Validate

Input: $\{Data-OK, H(M)\}$
SC receives $\{Data-OK, H(M)\}$ from DR;
if Σ *is 'traded' and T4 is valid* **then**
  if *On receive {Data-OK} or T4 expires* **then**
    DR's account balance -= $Price_M$;
    DO's account balance += $Price_M$;
    assert $\Sigma \equiv$ 'completed';
    send $\{Completed-Data\}$ to all nodes;
  end
end

[그림 11] 데이터 다운로드 및 검증 단계의 스마트 컨트랙트

바. 분쟁 해결(Dispute resolution request)단계

DR이 '데이터 다운로드 및 구매 확인 단계'에서 유효하지 않은 데이터를 획득했다면 분쟁 해결 단계로 진입한다. 여기서, 정직한 DR이 유효한 데이터를 획득했다면 프로토콜에 불평하지 않는 자연스러운 경우로 가정한다. 따라서 DR은 복원된 데이터 M의 유효성에 문제가 발생한 경우, $T4$ 이내에 SC와 A에게 분쟁 해결 요청을 위해 $\{Dispute-Data\}$를 각각 전송한다.

$$\{Dispute, ID_{DO}, ID_{DR}, ID_A, H(M), err-info, I_{SK}\}$$

I_{SK}가 의미하는 것은 DR이 데이터를 복원할 때 사용한 개인키 SK_{DR}을 DRC 알고리즘을 이용해 암호화한 값이다: $I_{SK} = DRC(SK_{DR}, PK_A, PK_{DO})$.

분쟁 해결 단계에서 SC는 DR에게 $\{Dispute-Data\}$를 수신했을 때와 A에게 $\{Resolution-Data\}$를 수신했을 때로 크게 2단계로 볼 수 있다. 자세한 과정은 [그림 9]에 의사 코드로 표현되었다. 먼저

DR로부터 $\{Dispute-Data\}$를 수신하면 SC는 분쟁 해결 요청이 $T4$ 이내인지, Σ가 'traded' 인지 확인한 후에 Σ를 'dispute' 로 설정한다. 위의 단계들과 마찬가지로 모든 노드들에게 $\{Dispute-Data\}$를 전송한다.

Algorithm 5: Dispute resolution request

```
/* Phase 1 - On receive {Dispute-Data} from DR */
Input: {Dispute – Data}
SC receives {Dispute – Data} from DR;
if T4 is before expiration and Σ is 'traded' then
    assert Σ ≡ 'dispute';
    store err-info, I_SK;
    send {Dispute – Data} to all nodes;
end
/* Phase 2 - On receive {Resolution-Data} from A */
Input: {Resolution – Data}
SC receives {Resolution – Data} from A;
if Σ is 'dispute' then
    assert Σ ≡ 'cancelled';
    if DO is the cheater then
        Forfeit the deposit of the DO;
        DR's account balance += Deposit_DR ;
    else if DR is the cheater then
        Forfeit the deposit of the DR;
        DO's account balance += Deposit_DO ;
    else // BN does not specify a cheater within the defined time
        Forfeit the deposit of the A;
        send {Cancelled – Data} to all nodes;
        DO's account balance += Deposit_DO ;
        DR's account balance += Deposit_DR ;
        Terminate;
    end
    DO's account balance -= Fee_BN ;
    A's account balance += Deposit_A + Fee_BN ;
    send {Cancelled – Data} to all nodes;
end
```

[그림 12] 분쟁 해결 단계의 스마트 컨트랙트

중재자 노드 A는 SC로부터 $\{Dispute-Data\}$를 받고 Σ가

'dispute' 면 err-info를 확인한 후 복원된 DR의 개인키와 블록체인에 게시된 PK_{DR}과 대응하는지, 그리고, 이 SK_{DR}을 통해 복원된 해시 값이 블록체인에 게시된 해시 값과 일치하는지, 서명이 유효한지를 검증함으로써 A의 분쟁 해결이 진행된다. 먼저 I_{SK}를 복호화하여 개인키 SK_{DR}을 획득한다: $SK_{DR} = DRC_{Dec}(I_{Stub}, SK_A, PK_{DO})$.

복원된 SK_{DR}이 블록체인에 게시된 PK_{DR}과 대응하는지 확인하고, 일치한다면 블록체인에 게시된 I_{Stub}을 DRC Recovery 알고리즘을 통해 $(eTK, Addr_{DO-M})$을 복원한다:

$$(eTK, Addr_{DO-M}) = DRC_{Rec}(I_{Stub}, PK_{DR}, SK_A).$$

외부 스토리지로부터 $(Package, \sigma_{pkg})$을 다운로드한 후 서명 σ_{pkg}를 검증한다. 복원된 SK_{DR}을 사용해 eTK를 TK''로 복원하고, 이를 이용해 $eStub$에서 $Stub''$을 획득한다. 서명 σ_{Stub} 까지 검증함으로써 서명에 대한 검증을 마치고, 마지막으로 위의 과정을 통해 획득한 $H(Stub \| Package)$와 $H(M)$의 해시 값을 검증한다. A가 진행하는 과정 중에서 하나라도 검증이 실패한 경우 DO를 부정행위자로 판별해 $\{Resolution-Data\}$에 "DO"를 부정행위자로 명시한 후에 SC에게 전송(입력)한다. 만일 그렇지 않다면 부정행위자를 "DR"로 명시한다.

$$\{Resolution, ID_{DO}, ID_{DR}, ID_A, H(M), (DO/DR)\}$$

SC가 중재자 노드로부터 $\{Resolution-Data\}$를 수신하면 제일 먼저 Σ가 'dispute' 인 것을 확인하고 'cancelled' 로 변경한다. $\{Resolution-Data\}$를 받고 SC가 할 수 있는 행위는 총 3가지로 나뉘는데 첫 번째는 DO가 부정행위자일 경우이다. DO가 부정행위자

로 판별나면 DO의 보증금을 몰수하고 DR의 보증금은 다시 DR에게 돌려준다.

두 번째는 DR이 부정행위자일 경우이다. 첫 번째와 마찬가지로 DR이 부정행위자로 판별되면 DR의 보증금을 몰수하고 DO에게는 보증금을 돌려준다. 두 가지 경우에서는 거래가 끝이 났기 때문에 DO의 계정 잔액에서 중재자 노드 A의 수수료만큼 차감하고 A에게 지불한다. 또한 A 역시 보증금을 냈기 때문에 보증금을 돌려받고, SC는 $\{Cancelled-Data\}$를 전송하고 거래를 종료한다.

마지막 경우는 중재자 노드 A가 미리 정의된 시간 내에 부정행위자가 누구인지 판별하지 않을 때이다. 이 경우에는 A의 보증금을 몰수하고, 미리 정의된 정책에 따라 DO와 DR에게 각각 분배, 각각의 보증금 역시 돌려준다. 그 후 모든 노드들에게 $\{Cancelled-Data\}$를 전송하고, 거래를 종료한다. 이 모든 과정이 끝나면 새로운 거래를 위해 DO는 새로운 키 쌍 (PK_{DO}', SK_{DO}')을 생성한 후 사전 준비 단계를 실시하게 된다.

사. 중재자 노드 선정

제안 프로토콜에서 중재자 노드 선정 과정은 DO가 '데이터 거래 공개' 단계에서 SC가 모든 노드들에게 $\{Sell-Data\}$를 전송하는 순간부터 타임스탬프 $T1$ 시간까지 이루어진다. 중재자 노드가 되고자 하는 네트워크 내의 노드들은 블록체인 노드를 관리하는 BNs-Blockchain에 자신의 정보를 포함한 트랜잭션을 게시한다.

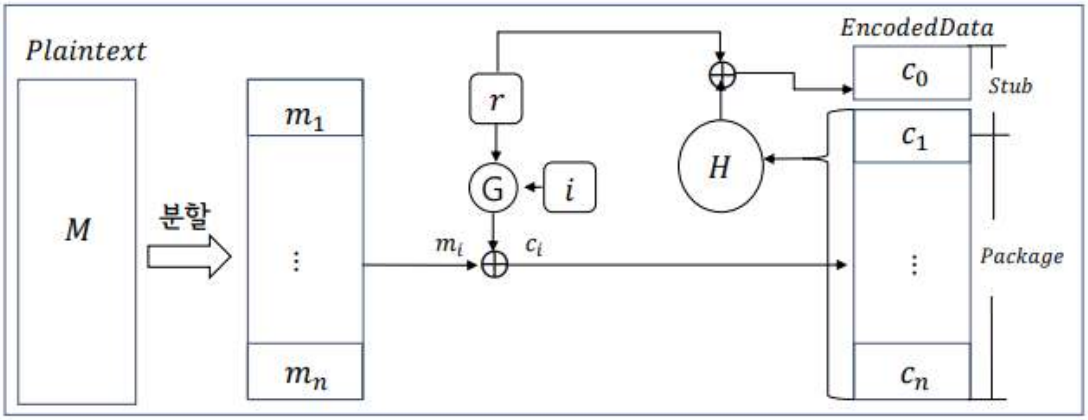
$$\{ID_A, ptime, PK_A, Deposit\}$$

여기서 $ptime$은 노드 i가 중재자로 참여하고자 하는 기간을 의미하며, $ptime$기간 동안 부정한 행위가 없었다면, 거래가 끝난 후 보증금은 다시 반환된다.

실제 제안 프로토콜에서 중재자 노드를 선정하는 방법은 다음과 같다. DO가 데이터 거래 공개 단계에서 메타데이터와 함께 (r, Q_{DO}^{r-1})을 포함해 블록체인에 게시하면 해당 트랜잭션에 관심이 있는 노드 i는 자체적으로 중재자 자격 증명서 $SIG_i(r, Q_{DO}^{r-1})$를 생성해 블록체인 네트워크에 브로드캐스팅한다. 그 후 DO는 미리 정의된 시간, 즉 T1 이내까지 브로드캐스팅된 중재자 자격 증명서들 중에서 $.H(SIG_i(r, Q_{DO}^{r-1})$가 가장 작은 값을 가진 노드를 해당 거래의 중재자 노드로 선정한다. 앞서 2장에서 설명한 것처럼 중재자 노드로 선정된 노드 i는 중재자 자격 증명서를 공개함으로써 해당 거래에 중재자 역할로 참여한다는 것을 밝힐 수 있다.

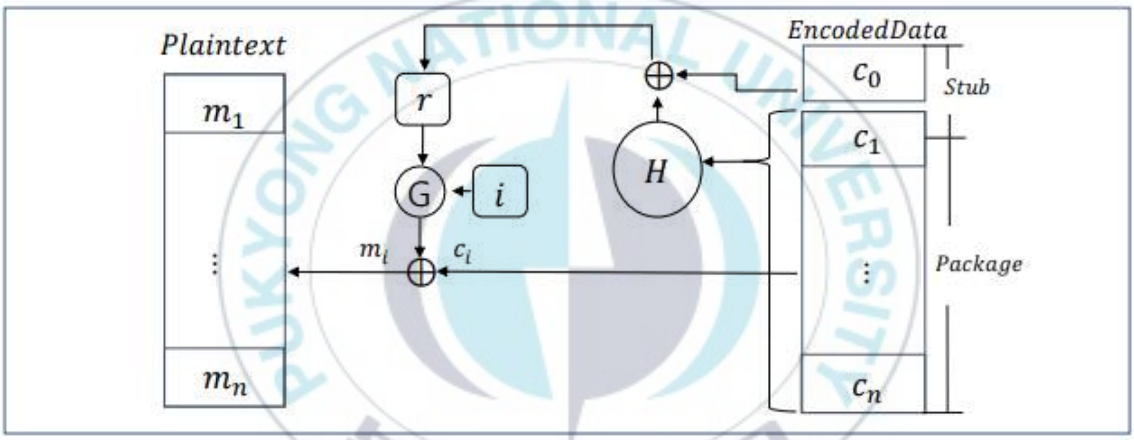
아. AONT 적용

제안 프로토콜에서 DO가 데이터 M을 전처리하기 위해 AONT를 다음과 같이 적용한다. 이 기법을 사용하기 위해 우리는 최소 256비트 출력을 지원하는 SHA-2, SHA-3와 같은 암호학적으로 안전한 해시 함수를 사용하는 것을 가정한다. [그림 10]을 통해 AONT의 암호화 과정을 알 수 있다. 먼저 랜덤한 256-bit의 r을 선택하고, 데이터 M을 블록 크기 혹은 bitrate 단위로 분할한다: $M = \{m_1, \cdots, m_n\}$. 분할된 블록 m_i를 함수 G를 이용해 XOR 연산을 시행한다: $c_i = m_i \oplus G(r \| i), i = 1, \ldots, n$. 여기서 함수 G는 암호학적으로 안전한 pseudo-random 함수이다. c_0는 r과 XOR 연산 값들의 해시 값을 XOR 연산을 시행하면 나오는 값으로 변환된 값 앞부분에 추가된다. 따라서 데이터 M은 AONT 처리 후 $Encoded\,Data = (c_0 \| c_1 \| \cdots \| c_n)$ 형태로 변환되었고, c_0와 c_1 일부가 $Stub$ 부분이 되고 나머지 부분이 $Package$가 된다. 일부분인 $Stub$은 일회용 키 TK를 이용해 AES 암호화 알고리즘으로 $eStub$으로 암호화된다.



[그림 13] AONT 암호화 과정

DR은 AONT 복호화 알고리즘을 통해 데이터 M으로 복호화 가능해야 하는 데 복호화 과정은 다음과 같다. 우선 AES 복호화 알고리즘을 통해 $(Stub \| H(SK_{DO}))$를 획득하고, 외부 스토리지에서 $(Package, \sigma_{pkg})$ 가져와 $(Stub \| Package)$쌍으로 구성한다. 그 후 $c_o \oplus H(c_1 \| \cdots \| c_n)$을 통해 랜덤 비트 r을 유추하고 암호화 과정에서 함수 G를 사용했던 연산을 입력 값을 바꾸어 계산한다: $m_i = c_i \oplus G(r||i), i = 1, \ldots, n$. 분할된 $\{m_1, \cdots, m_n\}$을 통해 데이터 M을 획득할 수 있게 된다. 이는 [그림 11]로 도식화되어 나타난다.



[그림 14] AONT 복호화 과정

IV. 블록체인 기반의 공정하고 안전한 데이터 거래 기법 구현 및 분석

본 장에서는 앞 장에서 제안한 모델의 실행과정을 실제로 구현한 코드와 실행 결과를 서술한다. 웹 페이지와의 연동을 통해 DO와 DR, 중재자 노드가 파라미터들을 입력할 수 있게 한다. 또한 스마트 컨트랙트의 구현으로 블록체인에 배포, 참여자들의 보증금과 데이터 가격 등 거래를 진행한다. 이더리움 프로젝트 구현을 위해 공식 문서와 다양한 오픈소스들을 참고하였다.

1. 구현 내용

이더리움 블록체인 네트워크를 기반으로 설계되었고 스마트 컨트랙트는 Solidity 언어로 작성되었다. 윈도우 상에서 React 프레임워크를 통해 입력받은 파라미터들을 변환시켜 스마트 컨트랙트를 작성한다. 작성된 스마트 컨트랙트는 3장에서 소개된 내용처럼 작동한다. 실제 동작을 확인하기 위해 Goerli 테스트 네트워크에서 거래를 진행하였다.

가. 개발환경

- CPU: Intel(R) Core(TM) i7-10700 CPU @ 2.90 GHZ
- 운영체제: Windows 10 Pro
- 메모리: 16GB RAM

- 하드디스크: 1 TB
- 개발 언어: Solidity 0.8.0, JavaScript XML(JSX)
- 오픈 소스

  Package: github.com/ethereum/go-ethereum

  Package: github.com/ethereum/solidity

  Package: github.com/adrianhajdin/project_web3.0

- 응용 애플리케이션

  Node.js 16.18.0

  Tailwindcss 3.18

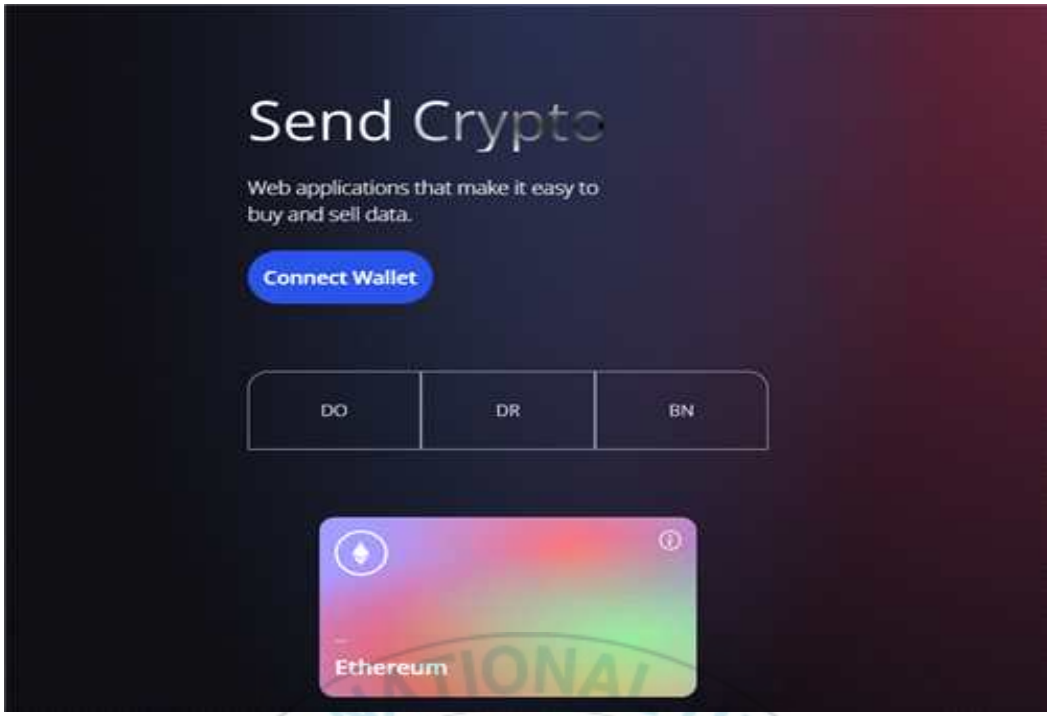
  Visual Studio Code 1.72.1

  React 18.2.0

  Hardhat 2.10.2

나. 제안 프로토콜 실제 실행 환경

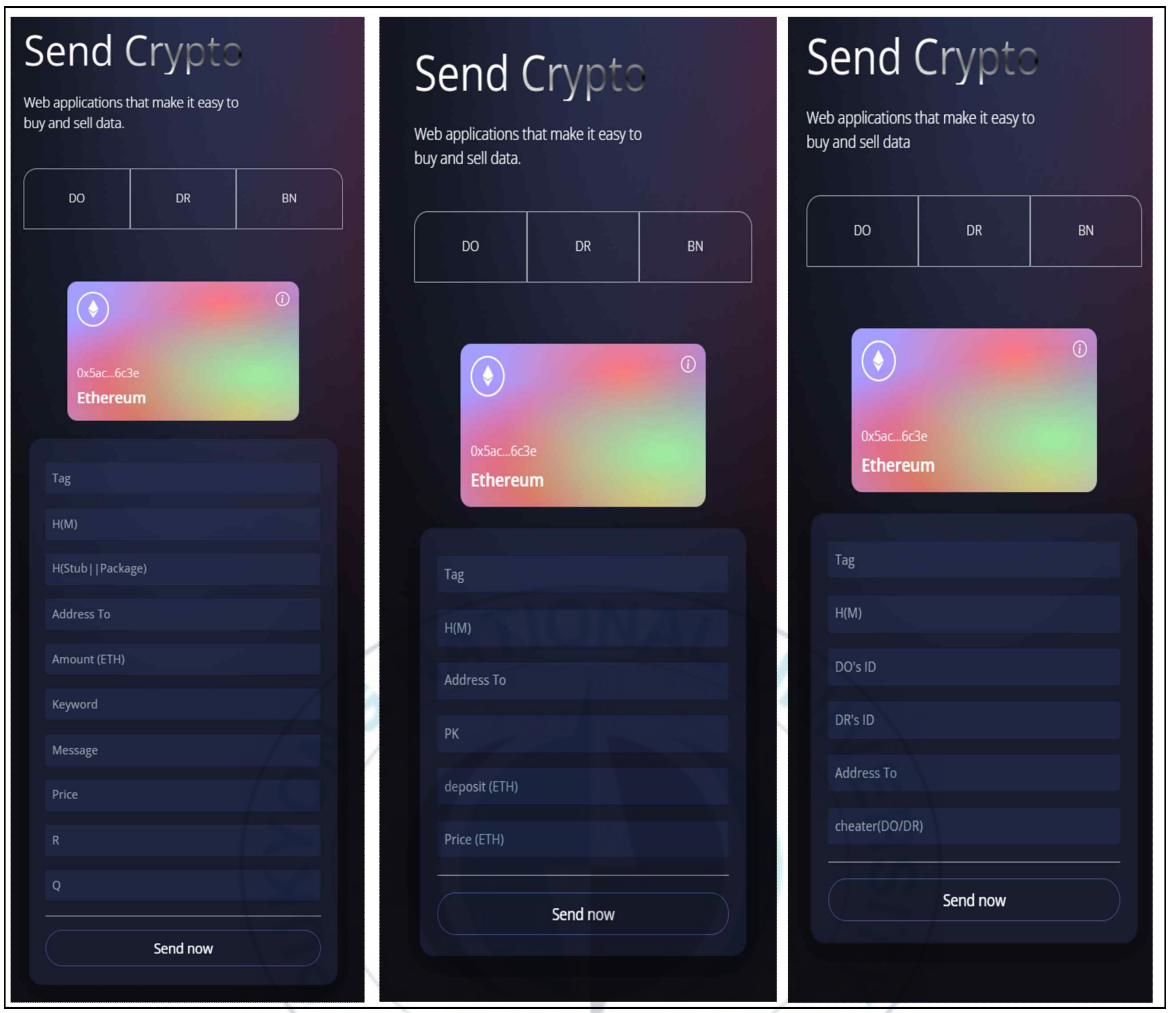
네트워크 내에 속해 있는 모든 개체들은 React를 이용해 스마트 컨트랙트를 작성할 수 있다. React 기반으로 생성된 웹 애플리케이션은 [그림 12]에서 보이듯이 먼저 자신의 계정과 연결되어있어야 한다. 계정 연결을 위한 코드는 [그림 13]에 나타나 있다. 먼저 메타마스크에 연결이 되어있는지 확인하기 위해 Ethereum 객체가 존재하는지 확인하고, 연결이 되어있지 않으면 'Please install metamask' 라는 에러 메시지를 보여준다. [그림 14]의 Connect Wallet 버튼을 누르면 메타마스크 연결 팝업창이 뜨고, 성공적으로 로그인한다면 [그림 14]의 페이지가 나오게 된다.



[그림 15] 웹 애플리케이션의 초기 로그인 화면

```
const checkIfWalletIsConnected = async () => {

    try {
        if(!ethereum) return alert("Please install metamask");

        const accounts = await ethereum.request({method: 'eth_accounts'});

        if(accounts.length) {
            setCurrentAccount(accounts[0]);

            getAllTransactions();
        }
        else{
            console.log('No accounts found');
        }
    } catch (error) {
        console.log(error);

    }


}

const connectWallet = async () => {
    try {
        if(!ethereum) return alert("Please install metamask");

        const accounts = await ethereum.request({method: 'eth_requestAccounts', });

        setCurrentAccount(accounts[0]);
        window.location.reload();

    } catch (error) {
        console.log(error);

        throw new Error("No ethereum object.")
    }
}
```

[그림 16] 계정 연결을 위한 코드



[그림 17] 거래 파라미터 입력받는 창(DO, DR, BN)

이 페이지는 DO, DR, BN 버전으로 나누어져 있어 자신이 맡은 역할에 따라 파라미터들을 작성할 수 있다. 각각 차례대로 DO의 입력창, DR의 입력창, 해당 거래의 중재자 노드의 입력창이다. 거래에 참여하는 참여자들은 웹 애플리케이션에 거래에 필요한 정보들은 [그림 15]의 구조체에 따라 입력 데이터를 받아오게 된다. 각 구조체는 계정 주소, 데이터, 가격, 타임스탬프 등이 요소로 속해있는데, 블록체인 네트워크 내에서 획득 가능한 보낸 개체의 주소, 타임스탬프는 입력받지 않고 자동으로 요소로 사용된다.

```
// SPDX-License-Identifier: UNLICENSED

pragma solidity ^0.8.0;

contract Transactions {

    uint256 transactionCount;

    event Publish(string tag, string p1, string p2, address from, address receiver, uint amount, string message,
uint256 timestamp, string keyword, uint price, string r, string q);

  event Trade(string tag, string p1, address from, address receiver, uint deposit, string pk, uint256 timestamp, uint
price);

  event Despite(string tag, string p1, address adr2, address adr3, address from, address receiver, uint256 timestamp,
string cheater, uint256 Deposit_DO, uint256 Deposit_DR);



    struct DOStruct{
        string tag;
        string p1;  // H(M)
        string p2;  //H(Stub||Package)
        address sender;
        address receiver;
        uint amount;
        string message;
        uint256 timestamp;
        string keyword;
        uint price;
        string r;
        string q;
    }

      struct DRStruct{
        string tag;
        string p1;
        address sender;
        address receiver;
        uint deposit;
        string pk;
        uint256 timestamp;
        uint price;

    }

      struct BNStruct{
        string tag;
        string p1;
        address adr2; //DO의 주소
        address adr3;  //DR의 주소
        address sender;  // 중재자노드의 주소
        address receiver;  // 스마트 컨트렉트 주소
        uint256 timestamp;
        string cheater;
        uint256 Deposit_DO;
        uint256 Deposit_DR;
    }

}
```

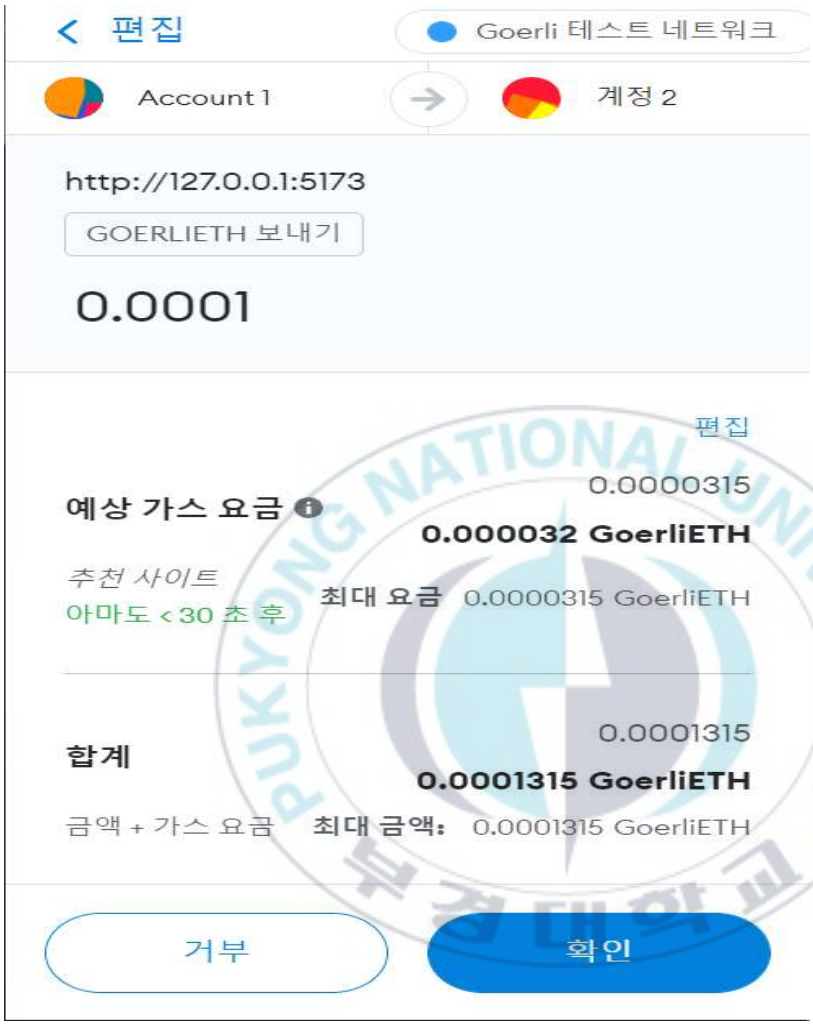
[그림 18] 각 참여 개체의 구조체

[그림 16]은 자동으로 입력된 요소들과 사용자가 직접 입력한 요소들을 이용해 Transaction을 생성, 전달하는 코드를 나타낸다. 이 코드는 Publish 단계에서 행해지는 내용을 담고 있으며 DO가 스마트 컨트랙트에게 $\{Sell-Data\}$를 입력하였다.

```
import React, { useEffect, useState } from 'react';
import { ethers } from 'ethers';
import { contractABI, contractAddress} from '../utils/constant';
export const TransactionContext = React.createContext();
const { ethereum } = window;

const getEthereumContract = () => {
    const provider = new ethers.providers.Web3Provider(ethereum);
    const signer = provider.getSigner();
    const TransactionContract = new ethers.Contract(contractAddress, contractABI, signer);

    return TransactionContract;
}

export const TransactionProvider = ({ children }) => {

   const sendTransaction = async () => {
        try {
            if(!ethereum) return alert("Please install metamask");

            const { tag, p1, p2, addressTo, amount, message, keyword,price, r, q } = formData;

            const transactionContract = getEthereumContract();
            const parsedAmount = ethers.utils.parseEther(amount);
            const parsedPrice = ethers.utils.parseEther(price);

            await ethereum.request({
             method: 'eth_sendTransaction',
             params: [{
                 from: currentAccount,
                 to: addressTo,
                 gas: '0x5208', //21,000 wei = 0.000021 eth
                 value: parsedAmount._hex, // 0.00001
             }]
            });

            const transactionHash = await transactionContract.addToBlockchain(tag, p1, p2, addressTo, parsedAmount,
message, keyword, parsedPrice, r, q);

            setIsLoading(true);
            console.log('Loading - ${transactionHash.hash}');
            await transactionHash.wait();

            setIsLoading(false);
            console.log('Success - ${transactionHash.hash}');

            const transactionCount = await transactionContract.getTransactionCount();

            setTransactionCount(transactionCount.toNumber());

        }
        catch (error) {
            console.log(error);

            throw new Error("No ethereum object.")

        }
    }
}
```

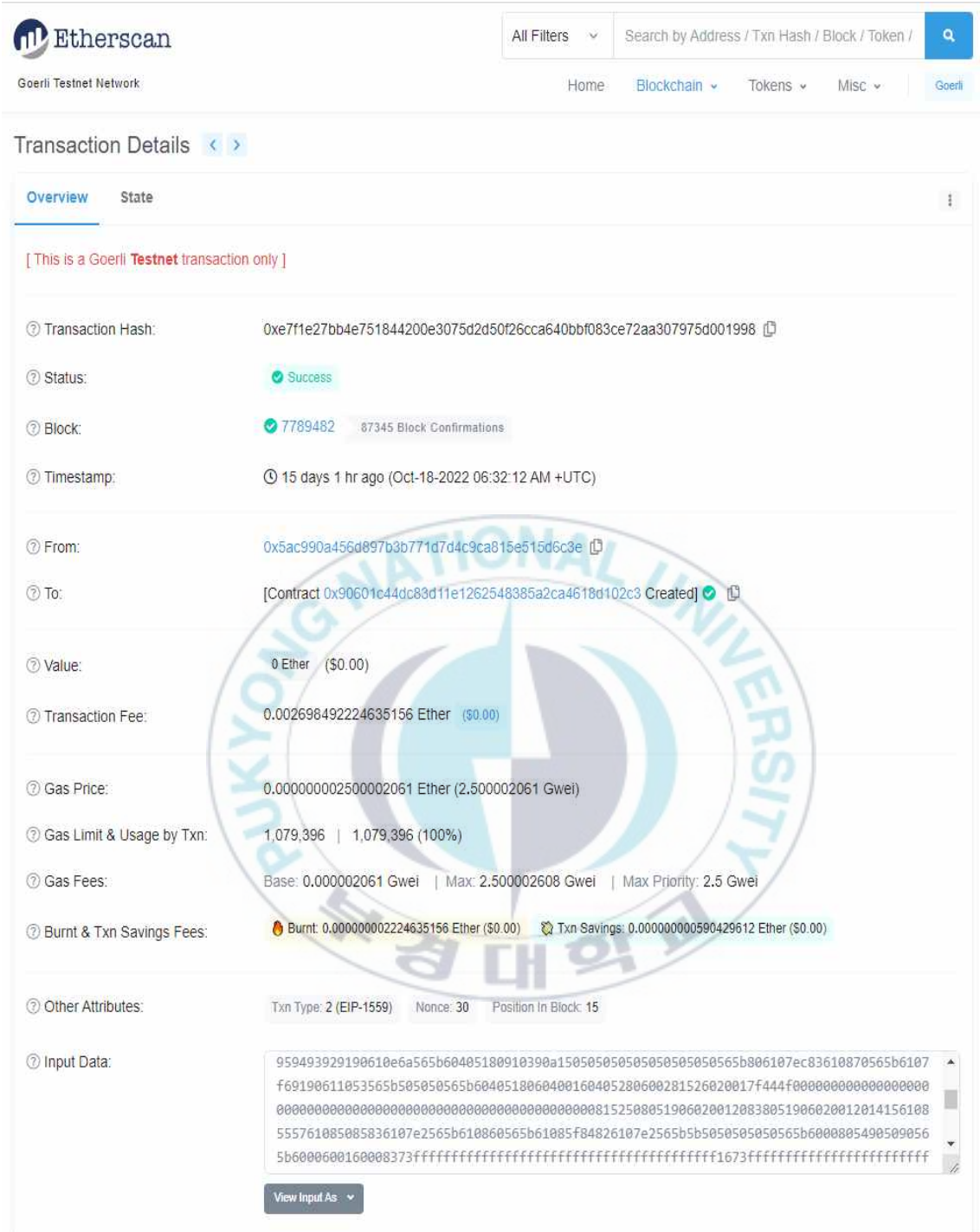
[그림 19] 트랜잭션을 위한 SendTransaction 함수

이 코드가 정상적으로 실행이 된다면, 메타마스크 팝업창을 통해 거래가 완료되었다고 [그림 17]처럼 나타난다.



[그림 20] 이더 송금 테스트 화면

또한 코드 내에서 구현된 것을 보면 성공적으로 SendTransaction이 실행된다면 트랜잭션의 해시 값이 터미널 창을 통해 출력된다. 이 해시 값을 Goreli 테스트 네트워크 트랜잭션 조회 사이트(Etherscan)에 입력하면 [그림 18]에서 보이듯이 자세한 트랜잭션의 내용을 확인할 수 있다.



[그림 21] 상세 트랜잭션 조회화면

DO가 성공적으로 Publish 단계에 트랜잭션을 배포하게 되면 DR은 자신의 페이지에서 Latest Transactions을 확인할 수 있다. [그림 19]를 보면 Latest Transactions에 DO가 입력한 데이터들을 볼 수 있고 자신이 구매하길 원하는 데이터가 있으면 스마트 컨트랙트를 통해 필요 정보들을 입력한다.

[그림 22] DR의 데이터 검색을 위한 페이지

데이터 구매 단계가 지나고 DR은 자신이 구매한 데이터가 유효한지 검증하게 된다. 만일 유효하다고 판단되면 거래는 종료되지만 유효하지 않다고 판단되면 중재자 노드가 부인 분쟁 해결 단계에 진입한다. 중재자 노드는 문제가 발생한 데이터를 다운받아 검증한 후에

어떤 참여 개체가 cheater인지 웹 애플리케이션에 작성한다. 그 후 아래 [그림 20]의 스마트 컨트랙트가 실행되어 cheater의 잔액 조회, 보증금 몰수가 이루어진 후 거래가 종료된다.

```
    function Balances(address _account) public view returns (uint balance){
        return balances[_account];
    }

    function minusBalance (address _adr, uint Deposit) public {

        Balances(_adr) - Deposit;
    }

    function penalty(address payable adr2, address payable adr3, string memory cheater, uint Deposit_DO, uint
Deposit_DR) public{
        if(keccak256(bytes(cheater)) == keccak256(bytes("DO")))
        minusBalance(adr2, Deposit_DO);
        else
        minusBalance(adr3, Deposit_DR);
    }
```

[그림 23] 정직하지 않은 노드 처벌을 위한 함수

2. 분석

제안 프로토콜은 3장에서 제시한 기본 목표를 모두 달성한다. 본 장에서 서술되는 분석은 제안 프로토콜이 데이터의 기밀성, 공정한 교환, 적시성을 만족하고 또한 두 번째 재판매 저항성도 만족한다는 내용을 담고 있다. 또한 제안 프로토콜은 기존의 다른 기법들과 달리 좀 더 높은 탈중앙성을 제공하는데, 이를 위해 비교 및 분석에 관해 서술한다.

가. 안전성 분석

본 항에서는 제안 프로토콜이 어떻게 목표를 달성했는지에 대해 분석한다. 총 5개의 Lemma와 1개의 Theorem으로 이루어져 있다.

[Lemma 1] 모든 참여 개체들이 정직하다는 것을 조건으로 제안 기법은 동기식 인증 네트워크 및 독립 실행형 모델에서 완전성을 만족한다.

(proof-sketch) 참여하는 DO와 DR 모두가 제안 프로토콜을 정직하게 따를 때 DO는 $Price_M$의 순이익을 얻고, DR은 유효한 데이터 M을 수신하므로 완전성을 충족한다.

[Lemma 2] 동기식 인증 모델 및 독립 실행형 모델에서 기반 암호 프리미티브들이 안전하다는 조건 하에 제안 프로토콜은 DO와 DR

중 한 당사자가 비적응적 P.P.T. 공격자에 의해 손상된 경우에도 공정성 요구 사항을 충족한다.

(proof-sketch) 공정성 요구 사항은 DR 공정성과 DO 공정성 두 가지를 모두 만족해야 한다.

(1) DR 공정성

DR 공정성은 DO가 악의적으로 행동하더라도 정직한 DR은 실질적으로 유효하게 획득한 데이터에 대해서만 지불한다는 것을 의미한다. 이것은 공격자가 정직한 DR에게 데이터를 제공하는 DO를 손상시키는 것으로 모델링할 수 있다. 거래 단계에서 공격자가 $\{Trade-Data\}$의 전달을 방해하거나 조작하려고 시도할 수 있으며, 이 경우 SC는 $\{Trade-Data\}$가 $T3$ 시간 이내에 도착하지 않는다면 거래를 취소시킨다. 따라서 공격자가 $\{Trade-Data\}$의 전달을 방해하는 것은 정상적인 거래가 이루어지지 않아 정직한 DR은 유효한 데이터를 획득하지 못하게 되므로 이에 대해 지불하지 않는다. 또한 공격자가 $\{Trade-Data\}$를 조작하는 것은 DR이 다운로드 & 검증 단계에서 획득 데이터의 유효성을 검증하게 되므로, 공격자가 해시 충돌을 발견하지 못하거나 서명의 위조가 불가능하고 블록체인에서 SC 실행을 조작할 수 없다면, DR의 공정성은 보장된다. 즉, 사용된 해시 함수와 서명 기법의 암호학적 메커니즘이 안전하고, 추가적으로 스마트 컨트랙트가 ideal functionality로 모델링 된다면, DR의 공정성을 break하는 확률은 negligible하다. 따라서 악의적인 DO에 대해 DR의 공정성은 보장된다.

(2) DO 공정성

DO 공정성은 정직한 DO는 DR에게 제공한 유효한 데이터에 대해 지불을 받는다는 것을 의미한다. 이것은 공격자가 DR을 손상시키는

것으로 모델링할 수 있다. 공격자는 DR이 지불하지 않고 유효한 데이터를 획득하게 함으로써 정직한 DO의 공정성을 break할 수 있다. 제안 프로토콜의 경우, 데이터 공개 단계에서 DO가 $\{Sell-Data\}$메시지의 전송을 통해 블록체인에 거래 데이터를 공개한다. 이 단계에서는 데이터 M에 대한 메타데이터만이 공개되므로, 실제 유효한 데이터를 획득할 수 없으며, 두 번째 단계인 데이터 요청 단계에서 DR이 $\{Buy-Data\}$를 통해 데이터 M에 대한 선지불을 한 후에 거래 단계에서 데이터M에 대한 획득 정보를 얻게 된다. 따라서 제안 기법의 거래 과정에서 DR은 지불하지 않고서 데이터 M을 획득하지 못한다. 공격자는 이 과정에서 $\{Trade-Data\}$를 통해 획득한 정보를 거래 요청하지 않은 DR에게 제공하려고 시도할 수도 있다. 이 경우, 거래 단계에서의 $\{Trade-Data\}$에서 $eStub$과 I_{Stub}을 획득할 수 있지만, I_{Stub}은 데이터 요청 단계에서 거래 요청을 한 DR의 공개키 PK_{DR}을 사용해 DRC 암호화 알고리즘이 적용되기 때문에 공격자는 복호화 알고리즘을 실시해 유효한 eTK와 외부 스토리지 주소를 얻을 수 없다(eTK 또한 PK_{DR}로 암호화되어 있다). $eStub$은 TK에 의해 암호화되어 있으므로, eTK를 복호화할 수 없다면 $Stub$을 획득할 수 없다. 따라서 제안 프로토콜에 사용된 기반 암호학적 메커니즘 즉, 공개키 암호, 대칭키 암호, DRC 암호기법이 공격자에 의해 break 될 수 없다면, DO의 공정성은 보장된다.

[Lemma 3] 적용된 중재자 노드 선정 기법이 안전하다는 조건 하에서 제안 기법은 공격자와 BN과의 공모 공격에 대해 안전하다.

(proof-sketch) 앞에서 기술했듯이, 공격자가 프로토콜 실행 이전에만 참여 개체를 붕괴시킬 수 있다는 것이 제안 기법의 기본 가정이

다. 따라서 공격자가 BN과의 공모를 통한 공격을 수행하기 위해서는 거래 프로토콜 실행 이전에 공모 공격을 수행할 BN을 선정해야 하고, 선택된 BN이 해당 데이터의 거래 프로토콜에서 중재자 노드로 선정되어야 한다. 공격자가 해당 거래 프로토콜 실행 이전에 미리 선정될 BN을 예측할 수 없으며, 따라서 해당 거래 과정에서 BN과의 공모 공격은 BN 선정을 위해 사용된 Algorand의 리더 선정 기법의 안전성에 의존한다. 즉, Algorand의 리더 선정 기법이 안전하다면, 제안 기법에서 공격자가 BN과의 공모 공격에서 성공 확률은 무시할 수 있다.

[Lemma 4] 동기식 인증 네트워크와 독립 실행형 모델에서 제안 기법은 비적응적 P.P.T. 공격자에 대해 기밀성을 만족한다.

(proof-sketch) 제안 기법의 원본 데이터 M은 외부로 아웃소싱되기 전에 AONT 기법을 통해 $(Stub, Package)$ 쌍으로 변환된다. $Package$는 외부 스토리지에 아웃소싱되며, 저장된 주소는 지불한 DR에게 DRC로 암호화된 주소가 전달된다. 또한 $Stub$ 부분은 TK를 적용한 대칭키 암호에 의해 $eStub$으로 암호화되며, TK는 비용을 지불한 DR의 일회용 공개키 PK_{DR}에 의해 eTK로 암호화된다. 그 후 DRC 프리미티브를 통해 암호화되어 블록체인에 공개된다. 따라서 P.P.T. 공격자가 데이터 M의 기밀성을 손상시키기 위해서는 사용된 DRC 프리미티브, eTK 암호화에 사용된 공개키 암호 알고리즘, $eStub$ 암호화에 사용된 대칭키 암호를 모두 break 해야 한다. 따라서 사용된 암호 메커니즘이 안전하다면, 비적응적 P.P.T. 공격자에 대해 제안 기법의 기밀성은 보장된다.

[Lemma 5] DO와 DR 중 한 개체가 정직하다면, 동기식 인증 네트워크와 독립 실행 모델에서 제안 기법은 적시성을 만족한다.

(proof-sketch) 적시성은 정직한 참여자는 항상 유한한 시간 내에 공정성을 보장한 상태에서 멈출 수 있는 프로토콜에서의 한 지점에 도달할 수 있다는 것을 의미한다. SC와 적어도 하나의 정직한 당사자를 가진 제안 기법에 대해 다음의 종료 케이스들을 상술한다:

(1) No abort: 모든 당사자가 정직하다면, 제안 기법은 다운로드 & 검증 단계에서 $\{Data-OK\}$가 수신되거나 $T4$가 만료된 후에 종료한다. 이때 DO와 DR은 원하는 것을 획득한 후에 종료된다는 것이 보장된다.

(2) 데이터 공개 단계에서 취소: 거래 데이터 공개 과정에서 타임스탬프 $T2$는 거래 요청에 대한 데드라인을 명시하며, $T2$가 만료되기 이전에 DR로부터 거래 요청받지 못한다면 거래는 취소된다. 이 지점에서 DO와 DR 모두에 대한 공정성은 보장된다: DO는 데이터 M을 제공하지 않았으며, DR은 거래에 대해 요청하지 않았다.

(3) 데이터 요청 단계에서 취소: 거래 요청 단계에서 타임스탬프 T3는 거래 단계에서 DO가 데이터 M에 대한 획득 정보 제공에 대한 데드라인을 명시한다. $T3$가 만료되었다는 것은 DO가 $\{Trade-Data\}$를 제공하지 않았다는 것이고, 따라서 SC는 거래를 취소한다. $T3$가 만료되면 DR은 지불한 비용을 돌려받음으로써 DR의 공정성을 보장하고, DO는 $\{Trade-Data\}$를 제공하지 않았기 때문에 역시 DO의 공정성이 보장된다.

(4) 거래 단계에서 취소: 거래 단계에서 타임스탬프 $T4$는 분쟁 해결 요청에 대한 데드라인을 명시한다. $T4$가 만료되었다는 것은 DR에 의한 분쟁 요청이 없었다는 것이고, 따라서 SC는 프로토콜을

정상 종료한다. 이 지점에서 두 당사자의 공정성은 보장된다.

(5) 분쟁 해결 과정에서 취소: 타임스탬프 $T4$ 이내에 DR에 의한 분쟁 해결 요청이 제기되면, 분쟁 해결 단계를 수행한다. 이 단계에서 지정된 시간 내에 중재자 노드 A로부터 $\{Resolution-Data\}$를 수신 받지 못한다면, SC는 A의 보증금을 몰수하여 DO와 DR에게 분배하고 거래를 취소한다. 따라서 이 지점에서 DO와 DR은 거래 취소에 대한 비용을 지불받기 때문에 공정성은 보장된다.

[Theorem 6] 기반이 되는 암호 프리미티브가 안전하고 블록체인/SC의 안전성이 보장된다면 동기식 인증 네트워크와 독립형 실행 모델에서 제안 프로토콜은 완전성, 공정성, 기밀성, 적시성을 만족한다.

(proof-sketch) Lemma 1, 2, 3, 4, 5에 의해 보장된다.

추가적으로 제안 기법은 '두 번째 재판매(Second reselling) 공격'에 대한 안전성을 지원한다. 두 번째 재판매란 악의적인 구매자가 거래 블록체인에서 데이터를 구매한 후 이익을 얻기 위해 데이터를 재판매하는 것을 의미한다. 제안 기법은 각 데이터의 해시 값이 블록체인에 게시되며, 판매자가 거래 공개 단계에서 거래 데이터를 공개할 때 블록체인/SC가 거래 데이터의 해시가 중복되는지 확인한다. 중복이 발견되면 원 데이터의 소유자인지 확인한 후, 판매자가 이전 거래에서 구매자였던 것이 확인되면 거래를 무효화할 수 있다. 따라서 제안 기법은 플랫폼 내에서 두 번째 재판매를 제거할 수 있다. 하지만, 공격자가 다른 방식이나 다른 플랫폼에서 데이터를 개인적으로 재판매하는 것은 막을 수 없다는 한계점이 있다.

나. 비교 분석

이 항에서는 제안 기법의 저장 및 계산 오버헤드를 분석하고, 블록체인 기반 공정한 교환을 지원하는 기존의 두 가지 연구와 비교한다.

(1) 제안 기법의 오버헤드 계산

먼저 제안 기법의 on-chain (블록체인) 상에서의 저장 복잡도를 살펴보면, 거래 과정의 각 단계에서 저장되는 정보는 다음과 같다:

- 데이터 공개 단계:

$$(ID_{DO}, H(M), H(Stub \| Package), Keywords_M, Terms_M, T1, Price_M, Fee_{BN}, Deposit_{DO}, PK_{DO}, T2)$$

- 데이터 구매 요청 단계: $(ID_{DR}, Deposit_{DR}, PK_{DR}, T3)$
- 거래 단계: $(ID_A, eStub, H(eStub), \sigma_{Stub}, I_{Stub}, T4)$
- 분쟁 해결 단계: $(err-info, I_{SK})$

저장 복잡도를 좀 더 구체적으로 살펴보았을 때, 제안 프로토콜이 128-비트 안전성을 가지는 암호학적 프리미티브의 키 길이를 가정하면 128-비트 대칭 암호, 256-비트 해시 함수, 그리고 256-비트 타원 곡선 페어링(elliptic curve pairings)의 비트 길이를 선택할 수 있다. 이 경우, PK와 SK는 각각 256-비트 길이, 서명의 길이는 ECDSA인 경우 512-비트, BLS 서명인 경우 256-비트 길이로 볼 수 있다[22]. 또한 128-비트 대칭 키를 평문 메시지로 입력한 비대칭 암호의 출력 암호문 길이는 512-비트, 256-비트 평문 데이터를 입력으로 하는 DRC 암호기법의 출력 암호문은 1024-비트이다. 160-비트 ID, 가격과 타임스탬프 길이는 64-비트, Addr를 128-

비트로 가정하면, 공개 단계에서 저장 복잡도는 (1248-비트 + 메타데이터 길이)이다. 따라서 메타데이터와 $err-info$를 제외하면 분쟁 해결 단계까지 진행된다고 했을 때 대략 1Kbytes 이하(5408-비트 길이)의 저장이 필요하므로 on-chain 상에서 저장 오버헤드는 그다지 높지 않은 것을 알 수 있다.

또한 기존 기법들과 달리 제안 기법의 기본 설계 원칙은 스마트 컨트랙트에서 복잡한 연산을 줄이는 것이다. 이더리움과 같은 블록체인 플랫폼의 스마트 컨트랙트가 튜링 완전성을 지원하지만, 여전히 스마트 컨트랙트에서 영지식 증명과 같은 복잡한 암호학적 연산을 처리하는 것은 처리 시간과 비용관점에서 많은 부담이 된다. 따라서 제안 기법의 스마트 컨트랙트에서는 비교, 할당, 저장, 가감산 연산만을 수행하도록 설계되었다. 복잡한 암호학적 연산은 참여 개체들이 로컬로 수행한다.

각 참여 개체의 암호학적 계산 오버헤드를 살펴보면 다음과 같다 (AONT 적용을 위해 블록을 n개의 블록으로 분할한다고 가정한다):

- DO의 관점
  - 준비 단계: $(n+1)*H$ 연산, $1*symEnc$, $1*Sign$
  - 데이터 공개 단계: $2*H$
  - 데이터 거래 단계: $1*AsymEnc$, $1*Sign$, $1*DRC_{Enc}$, $1*H$
- DR의 관점
  - 다운로드 단계: $1*DRC_{Dec}$, $2*Verify$, $1*AsymDec$, $1*symDec$, $(n+1)*H$, $2*H$
  - 분쟁 단계: $1*DRC_{Enc}$
- BN의 관점
  - 분쟁 단계: $1*DRC_{Dec}$, $1*DRC_{Rec}$, $2*Verify$, $1*AsymDec$,

$1*symDec$, $1*AONT$, $2*H$

CCA-secure DRC 프리미티브를 기본 연산 관점에서 계산량을 보면, 다음과 같다:

- DRC_{Enc}: 2*exp, 1*pairing, 3*H
- DRC_{Dec}: 1*exp, 1*pairing, 3*H
- DRC_{Rec}: 1*exp, 1*pairing, 3*H

또한 AONT 연산을 고려할 때, 기본 해시 함수의 연산으로 환산해 보면, 평문 메시지 M이 n개의 블록으로 구성된 경우, $(n+1)$번의 해시 연산으로 볼 수 있다.

(2) 기존 연구의 오버헤드 계산

기존 기법으로 Ahmad 등의 논문은 CP-ABE 기법과 zk-SNARKs 기반의 공정한 교환 기법을 제안했다. 이 기법에서는 CP-ABE에 사용될 키 생성을 위해 KDC(Key Distribute Center)라는 신뢰 개체 즉, TTP가 필요하게 되는데 이는 탈중앙성을 저해하는 요소로 작용된다. 또한 128-비트 안전성을 가정할 때 zk-SNARKs에서 Gate count가 2^{10}으로 가정한다면, 증명 키(pk)는 약 311.4KBytes, 검증 키(vk)는 3.6 KBytes의 크기를 가진다[23]. 따라서 이 기법에서는 총 630 KBytes의 키 쌍의 관리 오버헤드가 필요하다. zk-SNARKs의 키 생성은 비용이 많이 들고(circuit의 런타임에서 준선형), 각 circuit에 대해 키 생성을 매번 새로 하게 된다면 계산량적으로 상당히 큰 비용을 초래한다. 또한, 판매자와 구매자 간의 오프체인 상에서 zk-SNARKs 기반의 Proof와 Verify 연산을 실행해야 하고, 스마트 컨트랙트는 Withdraw 단계에서 상대적으로 높은 비용의 서명 검증 연산을 실행해야 한다.

[표 2] 제안 프로토콜과 기존 공정한 교환 연구의 비교

	Ahmad 등의 논문[17]	Li 등의 논문[18]	제안 프로토콜
암호학적 프리미티브	CP-ABE, zk-SNARKs	PCE	AONT, DRC
기밀성	정상적인 거래 과정에서는 기밀성이 보장되지만, 분쟁 과정에서 약한 기밀성 성질을 가진다.	정상적인 거래 과정에서는 기밀성이 보장되지만, 분쟁 과정에서 약한 기밀성 성질을 가진다.	거래 전체 과정에서 보장된다.
공정한 교환	보장된다.	보장된다.	보장된다.
적시성	보장된다.	보장된다.	보장된다.
탈중앙성	CP-ABE의 적용으로 KDC라는 TTP의 개입이 존재한다.	완전히 신뢰되지 않은 채굴자에 과하게 의존하는 경향이 있다. 하지만 정직하지 않은 행위를 한 채굴자에게 패널티를 부여하는 정책이 따로 존재하지 않는다.	선정된 중재자 노드를 완전히 신뢰할 수 없지만, 선정 과정이 완전히 랜덤이고, 정직하지 않은 행위를 했을 때 패널티가 있다는 특징이 있다.

본격적으로 Ahmad 등의 기법의 오버헤드를 분석해보면, CP-ABE 프리미티브를 적용하므로 128-비트 안전성을 가정했을 때(256-비트+ Na*(256*2)-비트) 길이의 구매자의 속성키가 요구된다. 여기서 Na는 속성의 개수이다. 그리고 CP-ABE 암호문의 길이는(256 + 256 + Na *(256*2))이다. 추가적으로 zk-SNARKs의 proof 길이는 288 bytes (2304 bits)이다. On-chain(스마트 컨트랙트) 상에서 요구되는 저장 정보는 $(ID, metadata, ID_M(=H(M)), Addr, AccessPolicy, CT, proof, fee)$ 이다. 이는 속성의 개수를 10개 정도로 가정했을 때 메타데이터와 Access Policy를 제외하면 8544-비트이다.

Ahmad 등의 기법의 문제점은 Withdraw 단계에서 비밀 값인 γ가 블록체인에 공개된다는 것이다. 이것은 거래 과정에서 공개된 정보를 통해 비용을 지불하지 않은 사용자가 데이터 획득이 가능하다는 의미이다. 또한 CP-ABE 기법의 적용하므로 Access policy를 만족하는 또 다른 사용자가 블록체인을 통해 공개된 $Addr$와 CT를 획득하고, Withdraw 과정에서 공개된 비밀 값 γ을 알게 되면 데이터 비용 fee을 지불하지 않고 데이터를 획득하는 것이 다음과 같이 가능하다: 부정한 구매자가 off-chain 공정한 교환 과정에서 C^*을 획득한 후, 프로토콜을 중단한다. 판매자는 프로토콜이 중단되었으므로, 해당 거래에서 비밀 값 γ을 전달하지 않는다. 하지만 부정한 구매자는 다른 정상적인 교환에서 공개된 비밀 값 γ을 사용하여 레코드를 획득할 수 있다. 이를 방지하기 위해 판매자는 각각의 교환에 대해 매번 새로운 비밀 값 γ를 사용하는 방안도 있지만 매번 C^*을 재계산해야 하는 오버헤드가 부가된다는 단점이 생긴다.

Li 등은 [17] 의 연구와는 다른 블록체인 기반의 공정한 데이터

거래 플랫폼을 제안하고 있다. 이 기법에서는 PCE(Plaintext Checkable Encryption) 프리미티브를 적용하고 있다. 제안 기법에서 분쟁이 발생하는 경우, 스마트 컨트랙트가 계산 복잡도가 높은 PCE_{chk} 연산과 서명 검증 연산을 on-chain 상에서 수행해야 한다. PCE 프리미티브의 복잡도는 다음과 같다:

- PCE_{Enc}: (256*4)-비트 길이, (4*exp + 2*mul) 계산 오버헤드
- PCE_{Dec}: (256*2)-비트 길이, (4*pairing + 1*exp + 1*div) 계산 오버헤드
- PCE_{Rec}: (256*3)-비트 길이, (4*pairing + 1*div) 계산 오버헤드

이 기법의 on-chain 상에서 저장되는 정보는 $(metadata, H_{root}, pk_{sign}, T_1, T_2, T_3, price, pk_{pce}, CT_{pce}, k_d, E_i, \sigma_i)$이다. 이를 128-비트 안전성을 가정했을 때, 메타데이터를 제외하면 대략적으로 2816-비트이다. 추가적으로 구체적으로 명시되어 있지 않은 ID의 크기도 고려한다면 대략 3136-비트이다.

Li 등의 기법은 공정한 교환을 위해 채굴자를 의존한다. 블록체인에서 채굴자는 완전 신뢰 개체가 아니며, 중재자 역할을 하는 채굴자들이 부정한 행위를 하는 것을 고려하지 않고 있다. 단순히 채굴자들이 제시된 증거들로 스마트 컨트랙트를 실행하여 검증하는 것을 가정하고 있으며, 공정한 거래를 중재한 채굴자에 대한 보상이 명시되어 있지 않으므로, 채굴자들을 거래 중재에 참여하게 할 인센티브가 부족하다. 또한 부정한 채굴자에 대한 처벌도 명시되어 있지 않아 공정하게 분쟁을 해결해야 하는 책임도 약하다. 또 다른 문제점은 분쟁 해

결 과정에서 비밀키인 k_D와 데이터의 암호문 E_i가 공개되는 것이다. 즉, 분쟁이 발생했을 때 다른 사용자들도 k_D와 E_i에 접근할 수 있게 된다. 부정한 구매자가 의도적으로 분쟁 과정을 실행하게 되면, 다른 사용자들은 비용의 지불 없이 데이터를 획득할 수 있다는 문제가 있다.

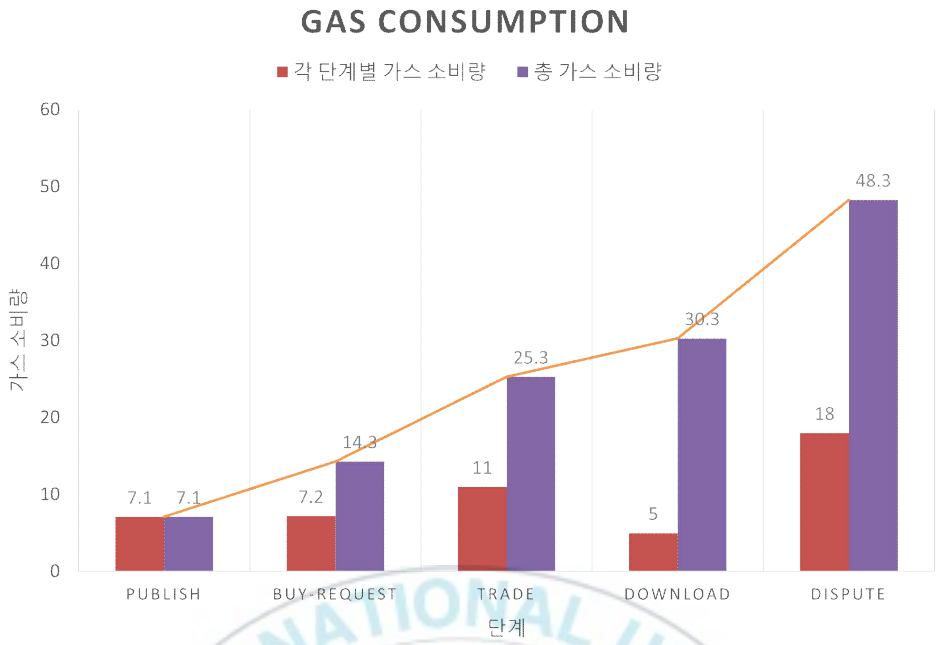
본 논문에 제안된 프로토콜은 위의 두 가지 기존 연구와는 달리, 분쟁 과정에서 데이터 획득을 위한 데이터 암호화 키가 직접적으로 노출되지 않는다. 거래 과정에 참여할 중재자 노드가 예측 불가능하게 선택되어 중재자 노드의 공개키를 적용하여 DRC에 의해 암호화되어 공개되므로 다른 사용자들은 이를 복호화할 수 없으므로, 분쟁 과정에서의 공개 정보를 통해 데이터를 획득할 수 없다. 또한 중재자 노드에게 공정한 거래 참여에 대한 보상 수수료를 지불함으로써 블록체인 내에 합의 노드들의 참여 동기를 부여하며, 또한 중재자 노드가 부정하게 행동한 경우, 참여 보증금을 몰수하는 처벌을 통해 공정하게 행동하도록 유도한다. 기존의 연구와 제안 기법과의 비교 분석을 [표 2]를 통해 제시하였다

[표 3] 제안 프로토콜과 기존 공정한 교환 연구의 비교

	Ahmad 등의 논문[17]	Li 등의 논문[18]	제안 프로토콜
온체인 스마트 컨트랙트 상에서 스토리지 오버헤드	$8{,}544-bit$ $(\approx 34words^{(\ddagger)})$ $(680{,}000\,gas)$	$3{,}136-bit$ $(\approx 13words)$ $(260{,}000\,gas)$	$5{,}408-bit$ $(\approx 22words)$ $(440{,}000\,gas)$
온체인 스마트 컨트랙트 상에서 복잡한 연산	서명 검증 연산이 필요	PCE_{chk}연산과 서명 검증 연산이 필요	복잡한 연산은 참여 노드들이 오프체인상에서 실시하고, 스마트 컨트랙트는 비교, 할당, 저장, 가감산 연산만 수행
블록체인 상에서 요구되는 가스 소비량	19×10^5	170×10^5	48×10^5

(‡) 이더리움 가상머신에서는 256비트 단위의 words를 사용한다.

제안 프로토콜은 가스 사용량을 측정하기 위해 이더리움 블록체인 상에서 구현되었다. 앞서 구현했던 스마트 컨트랙트를 통해 가스 사용량을 측정한 결과가 [그림 21]에 나타나 있다. 사전 준비 단계를 제외한 5단계를 측정하였고, 데이터 공개 단계에서 7.1×10^5, 데이터 요청 단계에서 7.2×10^5, 거래 단계에서 11×10^5, 다운로드 단계에서 5×10^5, 마지막으로 분쟁 단계에서 18×10^5 만큼 가스를 사용하며, 모든 단계를 계산했을 때 약 48×10^5 가스를 소비한다. 다른 두 기법과 가스 소비량을 비교했을 때 가장 높은 가스 소비량을 차지하는 SSTORE 연산을 기준으로 Ahmad 등의 기법은 약 680,000 가스를 소비하고 Li 등의 연구는 약 260,000 가스를 소비한다. 그에 비해 제안 프로토콜은 440,000 가스를 소비한다고 추정되고, 이를 on-chain 상에서 전체 가스 소비량을 비교했을 때, Ahmad의 연구는 19×10^5, Li의 연구에서는 170×10^5이다. 제안 프로토콜은 분쟁 상황이 발생하지 않으면 30×10^5가스를 소비하고 분쟁 상황 발생 시에는 48×10^5 가스를 소비하여, 기존의 다른 두 연구와 비교해 중간에 속한다고 볼 수 있지만 두 연구 모두 서명 검증 연산이 고려되지 않았고, 특히 Li 등의 연구에서 복잡한 연산인 PCE_{chk}연산이 스마트 컨트랙트에서 수행되어야 하기 때문에 스마트 컨트랙트에서 기본적인 비교, 할당, 저장, 가감산 연산만을 수행하는 제안 기법이 gas 비용관점에서 비교적 효율적임을 알 수 있다. [표 3]은 기존 연구들과 제안 프로토콜을 계산관점으로 비교 및 분석한 결과를 나타낸다.



[그림 24] 제안 프로토콜의 단계별 가스 소비량

V. 결 론

4차 산업 혁명 이후 빅데이터 시대로 전환되면서 본질적인 데이터와 데이터 소유권 관련해 많은 집중과 관심을 받고 있다. IoT 기기들로부터 생성되는 수많은 민감 데이터의 처리, 전달, 저장에 관해 많은 연구가 진행되고 있으며 데이터 공유/거래 시스템 역시 변화하고 있다. 기존의 중앙 집중화된 서버가 거래를 주관하거나 거래 데이터를 저장, 관리 또는 분쟁의 중재자로서 해야 할 역할을 하게 되면 중앙 집중 서버에 과도하게 신뢰하게 되고, 이는 중앙 서버가 공격지점이 되는 상황에서 데이터가 유출 및 변조가 될 수 있다. 또한 공격받거나 시스템 자체의 여러 요인들로 인해 서버 자체가 단일 실패 문제가 발생하며 성능 병목현상이 발생할 수 있다. 비공개 플랫폼은 안전하지 않은 채널에서 거래가 진행되고, 수차례의 협상으로 인해 많은 시간과 비용을 초래한다. 데이터 판매자와 구매자가 모두 정직하다는 시나리오만 존재한다면 문제가 없지만, 거래에 참여하는 노드들은 정직하지 않거나 호기심이 많기 때문에 두 거래자 사이의 거래는 공정하다고 볼 수 없고, 책임자가 따로 존재하지 않기 때문에 피해가 발생하였을 때 문제를 해결할 수 없다는 문제점이 있다. 이러한 문제를 해결하기 위해서 블록체인 기술을 도입한 부인방지 연구들이 증가하였다. 블록체인 기술은 P2P 방식을 채택하여 노드들이 거래 정보를 보관하기 때문에 거래의 위변조가 어렵고, 노드가 거래 정보를 처리하여 TTP의 개입을 피할 수 있다.

본 논문은 블록체인 기술과 암호학적 프리미티브들을 이용해 공정성과 탈중앙화를 지원하는 안전한 데이터 거래 기법 제안하고 구현한

내용을 분석했다. Algorand 논문에서 제안된 리더 선정 기법을 통해 완전히 랜덤하게 선정된 블록체인 중재자 노드는 프로토콜 실행 이전에만 공격자와 공모할 수 있지만 자기 자신을 제외한 누구도 중재자 노드가 어떤 노드인지 알 수 없으므로 사전 공모가 불가능하다. 이러한 방식으로 선정된 중재자 노드는 데이터 소유자와 데이터 요구자 사이에서 발생한 분쟁을 해결하면 수수료를 받고, 정직하지 않은 행위를 하였을 때 사전에 지불하였던 보증금을 몰수당하게 됨으로써 분산성이 보장되고, 공정한 교환이 실현된다. 데이터 소유자는 자신의 데이터에 AONT와 DRC 등의 암호화 알고리즘을 사용하였으며, 이는 거래에 참여한 당사자만이 복호화 가능하기 때문에 데이터에 대해 기밀성을 부여할 수 있다. 또한 전처리 된 데이터를 저장하는 방식으로 외부 분산화된 스토리지를 사용함으로써 블록체인 자체에 부담을 덜어줄 수 있으며, 거래의 안전성까지 부여하였다.

본 논문에서 제안된 프로토콜은 React 기반으로 구현된 웹 프론트엔드와 참여자가 서로 통신하고, 참여자가 작성한 파라미터를 기반으로 스마트 컨트랙트가 자동으로 작성되는 구조이다. 제안 프로토콜에서 스마트 컨트랙트 가스 소비량을 줄이기 위해 스마트 컨트랙트 상에서 비교, 할당, 가감산 등의 연산만 실행하고 암호화 같은 복잡한 연산은 참여 노드들이 오프체인에서 실행한다. 따라서 기존의 다른 연구들보다 가스 소비량이 작고 온 체인 상에서의 스토리지 오버헤드도 적당한 수치를 갖는다. 또한 제안 프로토콜은 데이터의 해시 값을 이용해 두 번째 재판매 저항성을 제공한다는 특징이 있다.

참고 문헌

[1] 과학기술정보통신부, 2020 데이터산업 현황조사 결과보고서, [Online]. Available: https://doc.msit.go.kr/SynapDocViewServer/viewer/doc.html?key=4a81cbc13f1b4f86959e958285c5e8d9&convType=img&convLocale=ko_KR&contextPath=/SynapDocViewServer

[2] Dai, W., Dai, C., Choo, K. K. R., Cui, C., Zou, D., & Jin, H. (2019). SDTE: A secure blockchain-based data trading ecosystem, IEEE Transactions on Information Forensics and Security, 15, 725-737.

[3] Bajoudah, S., Dong, C., & Missier, P. (2019). Toward a decentralized, trust-less marketplace for brokered iot data trading using blockchain. *IEEE international conference on blockchain (Blockchain). 339-346, 14-17 July 2019, Atlanta, USA.*

[4] ˇZiga Kodric, S. V., & Jelovcan, L. (2021). Securing edge-enabled smart healthcare systems with blockchain: A systematic literature review. Journal of Internet Services and Information Security (JISIS), 11, 19-32.

[5] Nakamoto, S. Bitcoin: A Peer-To-Peer Electronic Cash System. Manubot. 2019. Available online: https://git.dhimmel.com/bitcoin-whitepaper/

[6] 한국데이터산업진흥원, 2020 데이터 산업 백서, 통권 23호, 1-253.

[7] Zhou, J., & Gollmann, D. (1997). Evidence and non-repudiat

ion. Journal of Network and Computer Applications, 20, 267-281.

[8] Zhou, J., & Gollman, D. (1996). A fair non-repudiation protocol. *Proceedings 1996 IEEE Symposium on Security and Privacy. 55-61, Oakland, CA, USA.*

[9] Huang, Q., Yang, G., Wong, D. S., & Susilo, W. (2008). Ambiguous optimistic fair exchange. *14th International Conference on the Theory and Application of Cryptology and Information Security. 74-89, 7-11 December 2008, Melbourne, Australia.*

[10] Chen, J., & Micali, S., Algorand, [Online]. Available: https://arxiv.org/pdf/1607.01341v9.pdf.xxx

[11] Diament, T., Lee, H. K., Keromytis, A. D., & Yung, M. (2004). The dual receiver cryptosystem and its applications. *In Proceedings of the 11th ACM conference on Computer and communications security, 330-343, 25 October 2004,Washington, DC, USA.*

[12] Rivest, R. L. (1997). All-or-nothing encryption and the package transform. *4th International Workshop on Fast Software Encryption. 210-218, 20-22 January 1997, Haifa, Israel.*

[13] Ferrer-Gomila, J. L., Hinarejos, M. F., & Isern-Deya, A. P. (2019). A fair contract signing protocol with blockchain support. Electronic Commerce Research and Applications, 36, 100869.

[14] Huang, H., Li, K. C., & Chen, X. (2017). A fair three-party contract singing protocol based on blockchain. *9th International Symposium on Cyberspace Safety and Security. 72-85, 23-25 October 2017, Xi' an, China*.

[15] Z. Li, W. Lei, H. Hu, and W. Zhang. A blockchain-based communication non-repudiation system for conversational service. In Proc. of the 13th IEEE International Conference on Anti-counterfeiting, Security, and Identification (ASID), Xiamen, China, pages 6-10. IEEE, October 2019.

[16] Xu, Y., Ren, J., Wang, G., Zhang, C., Yang, J., & Zhang, Y. (2019). A blockchain-based nonrepudiation network computing service scheme for industrial IoT. IEEE Transactions on Industrial Informatics, 15, 3632-3641.

[17] Alsharif, A., & Nabil, M. (2020). A blockchain-based medical data marketplace with trustless fair exchange and access control. *In GLOBECOM 2020-2020 IEEE Global Communications Conference. 1-6, 7-11 December 2020, Taipei, Taiwan*.

[18] Li, Y. N., Feng, X., Xie, J., Feng, H., Guan, Z., & Wu, Q. (2020). A decentralized and secure blockchain platform for open fair data trading. Concurrency and Computation: Practice and Experience, 32, e5578.

[19] Kosba, A., Miller, A., Shi, E., Wen, Z., & Papamanthou, C. (2016). Hawk: The blockchain model of cryptography and privacy-preserving smart contracts. *In 2016 IEEE symposiu*

m on security and privacy (SP). 839-858, 22-26 May 20 16, San Jose, CA, USA.

[20] Gu, C., Zhu, Y., & Zhang, Y. (2005). An ID-based optimisti c fair signature exchange protocol from pairings. *In Internat ional Conference on Computational and Information Science. 9-16, 15-19 December 2005, Xi' an, China*.

[21] He, S., Lu, Y., Tang, Q., Wang, G., & Wu, C. Q. (2021). Fa ir peer-to-peer content delivery via blockchain. In Europe an Symposium on Research in Computer Security (pp. 348 -369). Springer, Cham.

[22] Boneh, D., Lynn, B., & Shacham, H. (2001). Short signatur es from the Weil pairing. *7th International conference on th e theory and application of cryptology and information secu rity. 514-532, December 2001, Gold Coast, Australia*.

[23] Ben-Sasson, E., Chiesa, A., Tromer, E., & Virza, M. (201 4). Succinct {Non-Interactive} zero knowledge for a von n eumann architecture. *In 23rd USENIX Security Symposium (USENIX Security 14). 781-796, 20-22 August 2014, San Diego, CA, USA*.

감사의 글

제 인생에서 다섯 번째 졸업을 가능하게 한 논문이 드디어 완성되었습니다. 저 혼자의 힘으로는 절대 불가능한 일들이 주변 많은 좋은 사람들의 도움으로 논문을 완성할 수 있었습니다.

우선 스승이라는 단어가 진정 어떠한 의미인지를 깨닫게 해주신 신상욱 교수님께 감사드립니다. 특히 2022년은 저에게 참 다사다난한 해였기에 교수님에게 많은 짐을 짓게 하고, 걱정도 많이 끼쳤습니다. 하지만 그럴 때마다 괜찮다, 건강이 우선이라고 말씀해주신 교수님께 감사합니다. 항상 감사하다고밖에 할 수 없었던 저 자신에게 실망하고 항상 죄송한 마음이었습니다. 교수님 덕분에 논문을 완성시키고 무사히 졸업할 수 있었기에 존경하는 마음을 담아 더 크고 멋있는 사람이 되도록 노력하겠습니다! 졸업 후에도 자주 연락드리고 찾아뵙겠습니다. 그리고 항상 좋은 강의와 저를 더 발전시키는 피드백을 쏟아내 주셨던 이경현 교수님과 신원 교수님께도 감사드립니다. 제가 5년 동안 연구실 생활하면서 좋은 사람들을 참 많이 만났습니다. 선배, 동기, 후배 할 것 없이 좋은 사람들과 친해질 수 있어 저에겐 연구실이 매우 뜻깊은 장소입니다. 저에게 할 수 있다고 격려해주신 선배님들, 믿고 응원해준 친구들과 동생들 정말 고마웠어요! 개인적으로 연락 전하면서 감사한 마음을 전달하도록 하겠습니다.

그리고 갑작스러운 대학원 진학에도 하고 싶은 공부 다 하게 해주신 부모님에게 감사합니다. 하나밖에 없는 딸이 그동안 속도 많이 썩이고, 속상한 일도 있으셨을 텐데 저를 믿고 기다려주셔서 고맙고 항상 존경합니다. 제 나름대로 열심히 준비해서 좋은 곳 취직해서 호강시켜드릴게요. 사랑해요!

마지막으로 제 옆을 항상 지켜주던 친구들에게 감사를 전하고 글을 마무리하도록 하겠습니다. 학교에서 집중 못하고 있는 친구를 위해 가끔 불러 수다도 떨고 좋은 음식, 좋은 풍경을 같이 즐겨 주고, 항상 같이 있어 줘서 고마웠습니다. 앞으로도 잘 지내자!

이 논문을 읽어주신 모든 사람이 항상 건강하고 행복하시길 기원하겠습니다.