https://doi.org/10.33778/kcsa.2020.20.4.177

동등한 권한을 가진 대표노드를 위한 능동적 비밀 분산을 이용한 비공개 블록 암호화 기법★

정 승 욱*

요 약

현재의 퍼블릭 블록체인은 누구나 원장의 내용을 볼 수 있도록 설계가 되어있다. 하지만 응용에 따라서 비밀 정보를 블록체인에 저장해야 하는 경우도 있으나 이에 대한 연구는 아직 미진하다. 본 논문에서는 DPoS(Delegated Proof of Stack) 합의 방식을 사용하는 블록체인을 대상으로 공개 블록과 비공개 블록의 두 계층으로 이루어진 블록체인을 제안하고 비공개 블록의 암호화를 위한 요구사항을 도출하였다. 도출된 암호화 요구사항을 만족하는 dealer없는 t-of-n threshold 암호화를 제안하였다. 또한, DPoS의 대표노드들은 가입과 탈퇴가 발생할 수 있어서, 대표노드의 가입과 탈퇴에 따라서 키 조각을 재분배하는 효율적인 방법을 제시하였다. 제안된 기법이 대표노드간의 공평성과 동일한 신뢰성을 만족하는 특징을 가진다.

Fair Private Block Encryption Protocol with Proactive Secret Sharing for Delegated Node of Public Blockchain

Seung Wook Jung*

ABSTRACT

In current public blockchain, any node can see every blocks, so that public blockchain provider transparent property. However, some application requires the confidential information to be stored in the block. Therefore, this paper proposes a multi-layer blockchain that have the public block layer and the private block for confidential information. This paper suggests the requirement for encryption of private block. Also, this paper shows the t-of-n threshold cryptosystem without dealer who is trusted third party. Moreover, the delegated node who has key information can be withdraw the delegated node group or a new delegated node can join in the delegated node group. Therefore, the paper proposes an efficient key information resharing scheme for withdraw and join. Finally proposed scheme satisfies the requirements for encryption and fairness.

Key words : Blockchain, Private Block, Threshold Cryptosystem, Proactive Secret Sharing

접수일(2020년 9월 28일), 게재확정일(2020년 10월 21일)

* 건양대학교/사이버보안공학과(corresponding author)

★ 본 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임 (No.2019-0-00411, 블록체인의 개인 콘텐츠 추적과 완전소멸수정을 위한 잊힐 권리 문제 해결)

1. 서 론

블록체인은 신뢰가 없는 인터넷에 신뢰를 제공해 주는 아주 유용한 기술이다[1]. 퍼블릭 블록체인의 특징은 한번 쓰인 블록은 변경할 수 없는 무결성, 불가역성 및 누구에게나 블록이 공개되는 투명성이 장점이다.

하지만 이런 장점 때문에 블록체인을 도입하고 싶어도 도입하지 못하는 다양한 응용이 존재한다. 예를 들어, 병무청의 업무 중 “병적별도관리대상자 명단”[2] 및 “병역사항신고 및 공개파일”[23] 등은 개인정보를 다루고 있으며 두 개인정보는 보유기간이 10년이며 시간이 지나면 삭제를 해야 한다. 하지만 블록체인은 무결성, 불가역성 때문에 해당 개인정보를 삭제하지 못해서 법령을 준수하지 못하는 문제가 있다. 또한, 해당 업무부서의 담당자와 시스템 관리자만이 접근하여 활용해야하는 기밀성의 요건도 가지고 있다.

이러한 상황에 대한 연구는 아직 없으며 본 논문에서 처음으로 다중-계층(Multi-Layer) 블록체인을 제안한다. 또한, 서로를 완전히 신뢰할 수 없는 노드들이 모여 있는 퍼블릭 블록체인, 특히 DPoS(Delegated Proof of Stack)을 사용하는 블록체인의 특징상 어떤 노드도 비밀값을 온전히 가지지 못하게 하고 몇몇 노드가 장애가 발생하거나 탈퇴하였을 때도 복호화가 문제가 없도록 신뢰할 수 있는 제3자 없는 Threshold 암호화 기법을 사용하는 방법을 제안한다. 또한, DPoS에서 노드가 탈퇴하거나 새로운 노드가 들어왔을 때 비밀값 재분산 기법을 제안한다. 게다가, 퍼블릭 블록체인의 비공개 블록을 위한 암호화 요구사항을 정리하였다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구를 살펴보고, 3장에서 다중-계층 블록체인의 필요성과 암호화 요구사항을 정리한다. 4장에서는 동등한 권한을 가진 대표노드를 위한 Threshold 암호화 프로토콜 및 대표 노드 탈퇴와 가입시 비밀값을 안전하게 교환하는 프로토콜을 제안한다. 5장에서는 보안성 및 성능을 분석하고 마지막으로 6장에서 결언을 맺는다.

2. 관련 연구

Sharmir가 비밀값 분산(secret sharing) 기법[8]을 발표한 후 후속 연구들이 진행되었다. 비밀값 분산과 threshold cryotosystem과의 차이점은 비밀값 분산은 서명이나 복호화에 참가한 참가자 어느 한곳에서 키가 복원되는 것이며 threshold crytosystem은 키가 복원 되지 않는 것이 특징이다.

후속 연구들은 다음의 방향으로 진행되었다.

1) 견고성 (reliability or robust)

2) 신뢰할 수 있는 dealer없는 시스템(no trust dealer)

3) 능동적 비밀분산 (proactive secret sharing)

4) 내부자 익명성 (insiders’ anonymity)

5) 효율성 향상

견고성은 서명이나 복호화에 참가한 참가자 중 하나가 잘못된 값을 만들어 냈을 때 이를 복원하는 것이다. 초기에는 전수조사를 통하여 누가 잘못된 값을 보냈는지 찾아냈었다[9]. 이후에 연구에서[10,11, 12]에서는 전수조사 없이 빠르게 잘못된 값을 찾는 연구가 있었다.

초기에서는 한 개의 신뢰할 수 있는 dealer가 있는 threshold cryotosystem을 연구하였으나 이후에는 신뢰할 수 있는 dealer 없는 threshold crytosystem[10, 13, 14]들이 개발되었다. 신뢰할 수 있는 dealer는 비밀값을 분산하기 전에 최초의 비밀키를 알고 있어 상당한 특권을 누릴 수 있다. 이에 본 연구처럼 공평성이 더 중요한 응용에서는 반드시 dealer 없는 시스템을 이용해야 한다.

임계치를 넘는 수 이상의 비밀값 조각들이 공격자의 손에 넘어가면 threshold crytosystem이 위험해진다. 이에 대한 대응으로 개인키와 공개키를 바꾸는 것이 있으나 매번 기관을 대표하는 공개키를 바꾸는 것은 바람직하지 않아 공개키를 바꾸지 않고 주기적

으로 비밀값 조각을 바꾸는 proactive threshold scheme[15, 16, 17, 18]들이 제안되었다. DPoS처럼 대표노드가 추가되거나 탈퇴되는 상황 및 서로를 전적으로 신뢰할 수 없는 시스템에서는 반드시 proactive threshold scheme이 고려되어야 한다.

외부에서는 누가 서명이나 복호화 등에 참여하였는지 모르나 내부에서는 누구나 알 수 있다. 비밀 투표와 같은 경우는 내부자들도 누가 참여하였는지 모르게 하는 익명성이 필요하여 [19]에서 연구되었으며 이와 반대로 추적성을 높이는 연구[20]도 진행되었다.

효율성은 모든 암호 프로토콜에서 필요로 하는 것이며 [21, 22] 등에서 효율성을 높이는 연구를 진행하였다. 블록체인 분야에서 블록을 암호화하는 기법에 대한 연구는 아직 없는 것으로 판단되며 추후에 연구들이 진행되어야 할 것이다.

3. 다중-계층(Multi-Layer) 블록체인

3.1 블록체인의 특징 및 합의 알고리즘

블록체인은 서로 신뢰할 수 없는 노드들의 집합인 네트워크에서 신뢰를 제공해 주는 기술이며 이에 블록체인은 신뢰기계라고도 부른다. 블록체인은 한번 원장에 기록된 내용을 바꿀 수 없는 무결성과 불가역성을 제공해 준다. 또한, 블록체인 네트워크에 참여한 누구든지 블록에 기록된 내용을 알 수 있는 투명성을 제공해 준다. 이러한 원장에 대해서 서로 신뢰할 수 없는 노드들이 합의를 통하여 신뢰할 수 있는 공통의 원장을 가지게 된다.

퍼블릭 블록체인의 합의 알고리즘은 대표적으로 PoW(Proof of Work), PoS(Proof of Stack), DPoS(Delegated Proof of Stack)이 있다[3].

작업 증명인 PoW는 채굴이라는 과정을 통하여 생성되는 블록에 대해서 합의를 하게 된다. 하지만 블록을 생성하는 채굴이라는 과정은 많은 전력 소모와 비용을 필요로 하는 단점이 있다.

이의 대안으로 지분 증명인 PoS가 나왔다. PoS는 해당 블록체인의 코인을 예치한 노드 중에 확률적으로 선출하여 선출된 노드가 블록을 생성하게 된다. 많은 코인을 예치한 노드가 선출될 확률이 높아지며 해당 코인을 가진 노드가 자신이 가진 코인의 가치를 훼손할 나쁜 일을 하지 않을 것이라는 것을 전제로 하여 블록생성을 신뢰하게 된다.

DPoS는 일정 수의 블록생산자라 불리는 대표노드를 뽑아 라운드 로빈 방식으로 블록을 만드는 방식이다. 코인 소유자가 자신이 직접 PoS를 하지 않고 대표노드에게 위임(delegate)하여 대표노드들이 블록을 만든 방식이다. DPoS는 PoS보다 빠르며 더 큰 규모의 트랜색션을 처리할 수 있다.

본 논문에서는 가장 최근에 나온 합의 알고리즘인 DPoS 블록체인을 가정하고 논문을 기술한다.

3.2 다중 계층 블록체인의 필요성

위에 설명한 어떤 노드이든지 블록내용을 확인할 수 있는 특징 때문에 기밀 정보를 저장하기에는 문제가 있다. 하지만 블록체인 응용서비스에 따라서 일부 비밀 정보를 저장할 필요가 있을 수 있다. 예를 들어, 병무청에서 운영하는 시스템에서 개인정보를 다루고 있어 기밀성이 필요한 정보와 일반에게 공개해야 하는 정보가 있다. 이와 같은 경우 공개 정보는 공개 블록에 저장하고 비공개 정보는 기밀성이 제공되는 비공개 블록에 저장하는 것이 합리적일 것이다. 따라서, 본 논문에서는 공개 블록과 비공개 블록을 가지는 다층 계층 (Multi-Layer) 블록체인을 제안한다.

공개 블록은 일반적인 블록체인의 블록과 동일하며 비공개 블록은 암호화를 하여 기밀성을 유지하도록 하였다. 여기서 블록을 생성하는 대표 노드들은 비공개 블록을 만들고 볼 수 있을 것이다. 하지만 일반 노드들은 암호화되어 내용을 볼 수 없다. 정리하자면 대표노드는 암호화된 비공개 블록을 만들고 복호화하여 내용을 볼 수 있으나 일반 노드는 해당 내용을 볼 수 없고, 비공개 블록에 들어있는 정보의 소유권을 가진 노드는 대표노드에 해당 내용을 복호화하여 보내 줄 것을 요청할 수 있다.

3.3 다중 계층 블록체인의 암호화 요구사항

비공개 블록이란 블록의 body가 암호화되고 블록의 header는 일반 블록체인과 동일하며 암호화된 body의 해쉬값이 header에 들어가게 된다. 다음은 다층 계층 블록체인의 암호화 요구사항이다.

1) 비독점적 암호화 키 : 비공개 블록을 생성하는 대표노드가 독점적으로 암호화 키를 가지게 되면 해당 대표노드가 대표노드에서 탈퇴나 자격 박탈을 당하였을 때 또는 해당 대표노드가 장애 등으로 동작을 하지 않을 때 복호화가 어려워진다. 따라서, t-of-n Threshold 암호화 기법을 이용하여 여러 대표노드들이 골고루 비밀 조각을 가지고 있으며 몇 개의 대표노드가 동작하지 않아도 정상적으로 복호화가 가능하도록 해야 한다.

2) 대표노드들의 동등한 신뢰성 : 비밀 조각 분배할 때 신뢰할 수 있는 제3자인 dealer없이 비밀 조각을 분배할 수 있어야한다. 즉 대표 노드 하나가 dealer가 되어 비밀 조각을 모두 안다면 해당 노드가 악의를 가졌을 때 블록체인 전체의 신뢰성에 문제가 발생한다. 또한, 대표 노드 하나가 분할되기 전의 온전한 비밀값을 알 수 없어야 한다. 따라서 신뢰할 수 있는 제3자 없는 t-of-n Threshold 암호화 기법을 고려해야 한다.

3) 다른 대표노드의 비밀 조각에 대한 무지 : 다른 대표노드의 비밀조각을 알 수 없어야 한다.

4) 대표노드 가입과 탈퇴시 효율적인 비밀값 재분산 : 대표노드가 탈퇴할 때 탈퇴한 여러 대표노드가 모여서 비밀키를 만들어 낼 수 없어야 하며, 가입시 효율적으로 비밀값을 업데이트 할 수 있어야 한다.

5) 검증된 암호화 알고리즘 사용 : AES(Advanced Encryption Standard)[6]처럼 널리 사용되는 안전한 암호화 알고리즘을 통하여 암복호화를 수행해야 한다.

DPoS에서 대표노드는 자신이 가지는 지분의 가치 하락을 고려하여 악의적인 행위를 하지 않을 것이라는 가정을 하고 있다. 그렇다고 대표노드를 무조건 신뢰할 수도 없다. 따라서, 대표노드들이 간에 서로를 적당히 신뢰(Semi-Trusted)하며 서로가 동등한 권한을 가지도록 블록체인 시스템을 설계해야 한다. 즉, 본 논문에서는 t-of-n Threashold 암호화 기법의 비밀키의 기밀성도 중요하지만 대표노드간의 공평성도 중요한 요구사항이다.

4. 동등한 권한을 가진 대표노드를 위한 비공개 블록 암호화 기법

4.1 AES-ElGamal에 기반한 그룹 복호화 기법

설정 (Set-UP)

G : 큰 소수 q의 위수를 가지는 군(group) = Z_q^*

H: G에서 AES Key길이의 비트열로 해쉬하는 함수

g : G의 생성자 (generator)

AES-ENC : AES 암호화 알고리즘

AES-DEC : AES 복호화 알고리즘

키 생성(Key Generation)

1. $SK = x \xleftarrow[R]{} G, PK = g^x \bmod q$
2. 다항식 $f(x) = \sum_{i=0}^{t} a_i z^i$ 를 선택, 단, $a_i \xleftarrow[R]{} G,\ f(0) = x$
3. $s_i = f(i) \bmod q,\ VK_i = g^{s_i}\ (1 \le i \le n)$ P_i에게 s_i를 안전한 채널로 전송한다.
4. $g, PK, (i, VK_i)_{1 \le i \le n}$ 를 공개

암호화(Encryption)

메시지 m의 암호문 E(m)을 생성하는 과정

1. $r \xleftarrow[R]{} G$
2. E(m) = $(g^r, AES-ENC_{H(g^{rx})}(m)) = (u, v)$

본 논문에서는 가장 일반적으로 사용하는 있는 블록 암호인 AES[6]에 기반한 그룹 복호화 기법을 이용한다. 해당 그룹 복호화는 [4]에 나와 있는 Hash-ElGamal[7] 그룹 복호화를 발전시킨 것이다. 해당 그룹 복호화는 t-of-n threshold 암호화에 기반을 하고 있다. AES-ElGamal 그룹 복호화는 다음과 같이 동작한다.

복호화 (Decryption)

암호문 E(b_m) = (u, v)를 복호화하는 과정

1. P_i는 모든 대표노드에 $u = g^r$를 전송하여 복호화를 요청
2. 각 P_j는 복호화 조각 $w_j = g^{rs_j}$를 P_i에게 전송
3. P_i는 오는 순서대로 $t-1$개의 w_j를 받으면 자신의 w_i를 포함하여 $g^{rx} = \prod_{j \in S} w_j^{\lambda_j}$를 계산

   여기서 $\lambda_j = \prod_{b \in S, j} \frac{j}{b-j}$는 Lagrange 계수

   또한 $S \subset [1, \cdots, n]$ $|S| = t$
4. P_i는 b_m = $AES-DEC_{H(g^{rx})}(v)$를 계산하여 복호화 한다

4.2 Dealer 없는 그룹 복호화 기법

DPoS를 합의 알고리즘으로 하는 블록체인에서 블록은 대표 노드가 생성하게 된다. 대표노드가 암호화된 비공개 블록을 생성할 필요가 있을 때 3.3절에서 전술한 암호화 요구사항을 만족하면서 비밀 조각과 비밀값을 생성해야 한다. 이를 위해서 본 논문에서는 [5]에 소개된 Dealer 없는 Threshold 암호화기법을 그룹 복호화에 맞게 수정하여 전체 프로토콜을 설계하였다.

그룹 복호화는 t-of-n threshold 암호화 기법에 기반하고 있다. 즉 n개의 대표노드 P = ($P_1, \cdots, P_n$)가 존재하고 t개의 대표노드가 협동하여 복호화를 할 수 있는 시스템이다. 또한, t-1까지의 해커에게 비밀 조각이 노출되어도 안전한 시스템이다.

대표노드들은 우선 다음의 설정 값을 서로 공유한다.

대표노드 P_l가 비공개 블록 생성 시점에 대표노드 P/P_l 에게 비밀값 생성을 위와 같이 요청하며, 자신도 비밀값 생성에 동참한다.

설정 (Set-UP)

G : 큰 소수 q의 위수를 가지는 군(group) = Z_q^*
H: G에서 AES Key길이의 비트열로 해쉬하는 함수
g : G의 생성자 (generator)
AES-ENC : AES 암호화 알고리즘
AES-DEC : AES 복호화 알고리즘

위의 비밀값 생성의 5번째 단계에서 s_{ii}는 자신이 비밀로 가지고 있으며 s_{ij}는 기밀성이 제공되는 안전한 채널로 서명을 붙여서 P_j에 전송한다.

다항식에 기반한 비밀값 생성

t-1차 다항식 $f(z)$를 생성하는 과정

1. P_i는 $x_i \xleftarrow[R]{} G, PK_i = g^{x_i} \bmod q$를 생성, PK_i를 다른 대표노드에 전송
2. n-1개의 PK_i를 수신한 대표노드는 공개값 $PK = \prod_{i=1}^{n} PK_i$를 생성한다.
3. P_i는 랜덤하게 t-1차 다항식 $f_i(z) = f_{i0} + f_{i1}z + \cdots + f_{i,t-1}z^{t-1}$를 생성한다. 여기서 $f_{j0} = x_i$임

   또한 $(f_{i0}, f_{i1}, \cdots + f_{i,t-1})$를 저장한다.
4. P_i는 $F_{ij} = g^{f_{ij}}$ $(1 \le j \le t-1)$를 계산하고 F_{ij} $(1 \le j \le t-1)$를 전송한다. 여기서 $F_{io} = PK_i$로 이미 알려져 있음
5. 모든 대표노드가 t-1개의 F_{ij}를 송신하고 난 후, P_i는 $s_{ij} = f_i(j)$를 P_j $(1 < j < n)$에게 안전한 채널을 통하여 전송한다.
6. P_i는 P_j가 보낸 s_{ji}가 맞는지 $g^{s_{ji}} = \prod_{l=0}^{t-1} (F_{jl})^{i^l}$ 식으로 확인
7. P_i는 5번째 단계에서 받은 모든 s_{ji}를 더하여 비밀 조각 $s_i = \sum_{j=1}^{n} s_{ji}$를 계산한다.

비밀값 생성의 결과로 다항식 $f(z)$는 G상에서 $f(z) = f_1(z) + \cdots + f_n(z)$가 만들어지고, $s_i = f(i), i = 1, \cdots, n$ 이 된다. 따라서, s_0는 $f(0) = x$의 비밀 조각이 된다.

위에 제시한 방식으로 대표노드들은 자신의 비밀 조각 s_i를 노출시키지 않고 비밀 조각을 특별한 권한을 가진 Dealer없이 나누어 가졌다. 즉 대표노드는 모두 동등한 입장에서 비밀 조각을 나누어 가졌다.

비공개 블록을 생성하기 위해서 비밀 값을 나누어 가진 후 P_l는 다음의 방식으로 비공개 블록을 암호화 한다.

암호화(Encryption)

m 번째 비공개 블록의 body b_m의 암호문 $E(b_m)$을 생성하는 과정

1. $r \xleftarrow[R]{} G$
2. P_l는 공개키 $PK = g^r$를 이용하여 다음과 같이 암호화한다.

$$E(b_m) = (g^r, AES-ENC_{H(g^{rx})}(b_m)) = (u, v)$$

복호화 (Decryption)

암호문 $E(b_m) = (u, v)$를 복호화하는 과정

1. P_i는 모든 대표노드에 $u = g^r$를 전송하여 복호화를 요청
2. 각 P_j는 복호화 조각 $w_j = g^{rs_j}$를 P_i에게 전송
3. P_i는 오는 순서대로 $t-1$개의 w_j를 받으면 자신의 w_i를 포함하여 $g^{rx} = \prod_{j \in S} w_j^{\lambda_j}$를 계산, 여기서 $\lambda_j = \prod_{b \in S \setminus j} \frac{j}{b-j}$는 Lagrange 계수

   또한 $S \subset [1, \cdots, n] \mid Svert = t$
4. P_i는 $b_m = AES-DEC_{H(g^{rx})}(v)$를 계산하여 복호화 한다

사용자가 m번째 비공개 블록의 복호화를 요청하면 다음 해당 블록 시간에 블록 생성을 담당하는 대표노드 P_i가 위의 복호화 절차를 통하여 복호화를 수행한다.

4.3 능동적 비밀 분산

DPoS를 적용한 블록체인에서 대표노드들이 대표노드에서 탈퇴하거나 새로운 대표노드가 가입할 수 있다. 따라서, 대표노드에서 탈퇴할 때와 새로운 대표노드가 가입했을 때 새롭게 비밀 분산 프로토콜을 수행해야 한다.

4.3.1 대표노드에서 탈퇴

t-of-n threshold 암호화 기법에서 t개의 대표노드가 탈퇴한 후 탈퇴한 대표노드들이 협동하여 4.2절의 복호화 방법을 이용하여 비공개 블록에 있는 내용을 복호화하여 볼 수 있게 된다. 이 때 대표노드에서 탈퇴한 노드는 블록체인의 원장에 접근할 수 있는 일반 노드일 수 있다. 일반 노드는 비공개 블록을 복호화할 수 없지만 대표노드에서 탈퇴한 일반 노드들이 t개 이상 되면 비공개 블록의 내용들을 볼 수 있게 된다.

이를 방지하기 위해서 주기적으로 능동적 비밀 분산 (Proactive secret sharing)을 적용하거나, 능동적 비밀 분산 프로토콜을 대표노드가 탈퇴할 때 마다 시행해야 한다. 본 논문에서는 대표노드가 탈퇴할 때 비밀 조각을 업데이트하는 것을 중심으로 설명한다. 이 능동적 비밀 분산 프로토콜은 [16]를 이용하였다. 만약 능동적 비밀 분산에서 주기를 하나의 대표노드가 탈퇴할 때까지로 생각하면 대표노드가 탈퇴할 때 비밀 분산 조각을 업데이트하는 것과 동일해진다.

본 논문에서는 새로운 주기(w)가 되었을 때 모든 대표노드들이 랜덤한 다항식을 만들어 공유하는 것이 아니라 대표노드 중 하나의 대표노드(P_i)만 랜덤한 다항식을 만들어 공유한다. 랜덤한 다항식을 만드는 대표노드는 라운드 로빈(Round Robin) 방식으로 선출된다.

만약 대표노드가 탈퇴가 아니고 단순히 주기적으로 업데이트를 한다면 아래의 대표노드 탈퇴시 비밀분산에서 0단계를 수행하지 않고 1단계의 $f_{i0}^{(w-1)} + f_{k0}^{(w-1)} + f_{i0}^{(w)}, \cdots, f_{i,t-1}^{(w-1)} + f_{k,t-1}^{(w-1)} + f_{i,t-1}^{(w)}$에서 t 첨자가 있는 것을 제거하면 된다.

0번째 주기에서는 4.2절에 설명한 다항식에 기반한 비밀값 생성 결과, 각 대표 노드의 다항식을 더한 최종 결과 다항식은 $f^{(0)}(z) = f_1(z) + f_2(z) + \cdots + f_n(z)$(단, $f^{(0)}(0) = x$)이다. 1번째 주기에는 최종 결과 다항식은 $f^{(1)}(z) = f^{(0)}(z) + f_i^{(1)}(z)$이며 w 번째 최종결과 다항식은 $f^{(w)}(z) = f^{(w-1)}(z) + f_i^{(w)}(z)$이다. 또한, 각 대표노드 i의 비밀 조각은 $f^{(w)}(i) = s_i$이다.

여기서, 비밀키인 x는 변함이 없으며 w주기에 랜덤한 다항식을 만드는 i는 $s_j^{(w)} = s_j^{(w-1)} + s_{ij}^{(w)}$ 식에서 $s_{ij}^{(w)}$ 를 알지만 $s_j^{(w)}$와 $s_j^{(w-1)}$를 알 수는 없어 보안이 유지된다.

대표노드 탈퇴시 비밀 분산

0. 만약 P_k 가 탈퇴한다면 자신의 다항식 계수 $(f_{k0}^{(w-1)}, f_{k1}^{(w-1)}, \cdots, f_{k,t-1}^{(w-1)})$ 를 P_i에게 전달함
1. P_i 는 w 주기에 랜덤하게 t-1차 다항식 $f_i^{(w)}(z) = f_{i0}^{(w)} + f_{i1}^{(w)} z + \cdots + f_{i,t-1}^{(w)} z^{t-1}$를 생성한다.

   여기서 $f_{i0}^{(w)} = 0$임

   $(f_{i0}^{(w-1)} + f_{k0}^{(w-1)} + f_{i0}^{(w)}, f_{i1}^{(w-1)} + f_{k1}^{(w-1)} + f_{i1}^{(w)}, \cdots, f_{i,t-1}^{(w-1)} + f_{k,t-1}^{(w-1)} + f_{i,t-1}^{(w)})$

   를 저장한다.
2. P_i는 $F_{il}^{(w)} = g^{f_{il}^{(w)}}$ $(1 \le l \le t-1)$를 계산하고 $F_{il}^{(w)}$ $(1 \le l \le t-1)$를 전송한다.
3. 모든 대표노드가 t-1개의 $F_{ij}^{(w)}$를 수신하고 난 후, P_i는 $s_{ij}^{(w)} = f_i^{(w)}(j)$를 $P_j (1 < j < n)$에게 안전한 채널을 통하여 전송한다.
4. P_j는 P_i가 보낸 $s_{ij}^{(w)}$가 맞는지 $g^{s_{ij}^{(w)}} = \prod_{l=0}^{t-1} (F_{il}^{(w)})^{j^l}$ 식으로 확인
5. P_j는 4번째 단계에서 받은 $s_{ij}^{(w)}$를 이용하여 비밀 조각 $s_j^{(w)} = s_j^{(w-1)} + s_{ij}^{(w)}$를 계산한다.

4.3.2 새로운 대표노드 가입

새로운 대표노드 P_i가 가입하였다고 가정한다. 새로운 대표노드가 들어왔으므로, 새로운 주기(w)가 시작되었다고 가정하자. P_i는 랜덤한 t-1차 다항식을 만들고 계수를 저장한다. 이 다항식을 이용하여 다른 대표노드 P_j에게 비밀값 $s_{ij}^{(w)} = f_i^{(w)}(j)$를 안전하게 전송하고 자신을 제외하고 검증값 $F_{ij}^{(w)}$ $(1 \le j \le t-1)$을 전송한다. P_j는 이를 이용하여 검증하고 맞다면, 기존의 비밀값에 새로운 비밀값 $s_{ij}^{(w)}$을 더하여 새로운 비밀값을 만든다.

기존의 대표노드 P_j는 자신이 저장한 다항식 계수 $(f_{i0}^{(w-1)}, f_{i1}^{(w-1)}, \cdots, f_{i,t-1}^{(w-1)})$로부터 자신의 t-1차 다항식을 복원하고 P_i를 위한 비밀값과 검증값을 만들어 전송한다. 검증값이 맞다면 P_i는 자신이 만든 다항식의 $s_{ii}^{(w)} = f_i^{(w)}(i)$ 과 P_j로부터 받은 모든 비밀값을 더하여 자신의 비밀값 $s_i^{(w)}$을 만든다.

새로운 대표노드 가입 시 비밀 분산

1. 새로운 대표노드 P_i 는 w 주기에 랜덤하게 t-1 차 다항식 $f_i^{(w)}(z) = f_{i0}^{(w)} + f_{i1}^{(w)} z + \cdots + f_{i,t-1}^{(w)} z^{t-1}$를 생성한다.

   여기서 $f_{j0}^{(w)} = 0$임

   $(f_{i0}^{(w)}, f_{i1}^{(w)}, \cdots, f_{i,t-1}^{(w)})$를 저장한다.
2. P_i는 $F_{ij}^{(w)} = g^{f_{ij}^{(w)}}$ $(1 \le j \le t-1)$를 계산하고 $F_{ij}^{(w)}$ $(1 \le j \le t-1)$를 전송한다.
3. 모든 대표노드가 t-1개의 $F_{ij}^{(w)}$를 송신하고 난 후, P_i는 $s_{ij}^{(w)} = f_i^{(w)}(j)$를 $P_j (1 < j < n)/i$에게 안전한 채널을 통하여 전송한다.
4. 기존 대표노드 P_j는 P_i가 보낸 $s_{ij}^{(w)}$가 맞는지 $g^{s_{ij}^{(w)}} = \prod_{l=0}^{t-1} (F_{il}^{(w)})^{j^l}$ 식으로 확인
5. P_j는 4번째 단계에서 받은 $s_{ij}^{(w)}$를 이용하여 비밀 조각 $s_j^{(w)} = s_j^{(w-1)} + s_{ij}^{(w)}$를 계산한다.
6. 기존의 대표노드 P_j는 저장된 $(f_{i0}^{(w-1)}, f_{i1}^{(w-1)}, \cdots, f_{i,t-1}^{(w-1)})$로부터

   $f_j^{(w-1)}(z) = f_{j0}^{(w-1)} + f_{j1}^{(w-1)} z + \cdots + f_{j,t-1}^{(w-1)} z^{t-1}$

   를 복원하고 새로 가입한 P_i를 위하여 $s_{ji}^{(w-1)} = f_j^{(w-1)}(i)$를 안전한 채널을 통하여 전송한다.
7. P_j는 P_i가 검증할 수 있도록 $F_{jl}^{(w-1)} = g^{f_{jl}^{(w-1)}}$ $(1 \le l \le t-1)$를 계산하고 $F_{jl}^{(w-1)}$ $(1 \le l \le t-1)$를 전송한다.
8. P_i는 P_j가 보낸 $s_{ji}^{(w-1)}$가 맞는지 $g^{s_{ji}^{(w-1)}} = \prod_{l=0}^{t-1} (F_{jl}^{(w-1)})^{i^l}$ 식으로 확인
9. P_i는 6번째 단계에서 받은 모든 $s_{ji}^{(w-1)}$과 자신의 비밀값 $s_{ii}^{(w)}$ 를 더하여 비밀 조각 $s_i^{(w)} = \sum_{j=1/i}^{n} s_{ji}^{(w-1)} + s_{ii}^{(w)}$를 계산한다.

기존의 대표노드들은 다항식의 계수를 저장하고 있고, 이를 이용하여 새로운 대표노드가 들어오면 다항식을 만들고 비밀값을 전송해 주는 방식이다. 여기서, 자신의 다항식 계수는 벡터 $(f_{i0}^{(w-1)}, f_{i1}^{(w-1)}, \cdots, f_{i,t-1}^{(w-1)})$로 생각할 수 있으며 $(f_{i0}^{(w-1)}, f_{i1}^{(w-1)}, \cdots, f_{i,t-1}^{(w-1)}) = (f_{i0}^{(0)}, f_{i1}^{(0)}, \cdots, f_{i,t-1}^{(0)}) + (f_{i0}^{(1)}, f_{i1}^{(1)}, \cdots, f_{i,t-1}^{(1)}) + \cdots + (f_{i0}^{(w-2)}, f_{i1}^{(w-2)}, \cdots, f_{i,t-1}^{(w-2)})$이며, 능동적 비밀 분산 프로토콜에서 u번째 주기가 자신이 다항식을 만드는 주기가 아니면 $(f_{i0}^{(u)}, f_{i1}^{(u)}, \cdots, f_{i,t-1}^{(u)}) = (0, 0, \cdots, 0)$이 된다.

5. 보안성 및 성능 분석

5.1 프로톨콜의 공평성

1) 비독점적 암호화 키 : 비공개 블록의 복호화는 대표노드 t개가 모이면 복호화할 수 있다. 따라서 블록을 생성한 대표노드 P_l이 탈퇴나 장애등으로 온라인 상태가 아니라도 복호화할 수 있어 비독적 암호화 키 조건을 만족한다.

2) 대표노드들의 동등한 신뢰성 : 비밀값 x 생성 과정에서 어떤 대표노드도 비밀값 x를 알지 못한다. 또한, 다른 대표노드가 가지고 있는 비밀 조각 s_i을 알 수 없다. 복호화하는 과정에서도 비밀값 x나 비밀 조각 s_i 을 알 수 없다. 따라서 어떤 대표노드도 다른 대표노드보다 특권을 가지고 있지 않아 동등한 신뢰성을 만족한다.

3) 다른 대표노드의 비밀 조각에 대한 무지 : 4장에서 소개된 프로토콜에서는 어떤 대표노드도 다른 노드의 s_i도 알지 못한다.

4) 대표노드 가입과 탈퇴시 효율적인 비밀값 재분산 : 대표노드 탈퇴 시 탈퇴한 대표노드의 비밀값을 무력화하기 위해서 4.3절에서 비밀값을 효과적으로 업데이트하였다.

5) 검증된 암호화 알고리즘 사용 : AES(Advanced Encryption Standard)를 통하여 암복호화를 수행한다.

5.2 프로톨콜의 성능

1) 비밀조각 분배 시 총 메시지 Byte 수

Security Parameter는 PK의 길이 l이라고 하자. 다항식에 기반한 비밀값 생성에서 1번째 단계에서 n개의 $l$byte 메시지가 발생하며 4번째 단계에서 $(t-1)\times n$개의 $l$byte메시지가 발생한다. 마지막으로 5번째 단계에서 n개의 메시지가 발생한다. 그러면 비밀 조각 분배시 총 메시지는 식 1)과 같이 표현할 수 있다.

$$(n+(k-1)\times n+n)\times l = O(n\times l) \quad 1)$$

2) 복호화시 메시지 Byte 수

복호화시에는 1단계에서 n개의 l byte 메시지가 발생한다. 2단계에서 n개의 l byte 메시지가 발생한다. 따라서 식 2)와 같이 총 메시지 byte 수를 나타낼 수 있다.

$$(n+n)\times l = O(n\times l) \quad 2)$$

3) 대표노드 탈퇴시 분배되는 총 메시지 Byte 수

대표노드 탈퇴 시 0단계에서 t-1개의 l byte 메시지가 발생한다. 2단계에서 $(t-1)$개의 l byte 메시지가 발생한다. 3단계에서 n개의 l byte가 발생한다. 따라서 식 3)와 같이 총 메시지 byte 수를 나타낼 수 있다.

$$(2t+n)\times l = O(n\times l) \quad 3)$$

4) 새로운 대표노드 참여시 분배되는 총 메시지 Byte 수

새로운 대표노드 참여시 2단계에서 t-1개의 l byte 메시지가 발생한다. 3단계에서 n개의 l byte 메시지가 발생한다. 5단계에서 n개의 l byte 메시지가 발생한다. 마지막으로 7단계에서 $(t-1)\times n$개의 l byte가 발생한다. 따라서 식 4)와 같이 총 메시지 byte 수를 나타낼 수 있다.

$$(n+n+(t-1)+n\times(t-1))\times l = O(n\times l) \quad 4)$$

6. 결 론

본 논문에서는 병무청의 공개정보를 관리하거나 개인정보를 노출없이 관리해야 하는 것 같이 공개 블록과 비공개 블록을 서비스해야 하는 블록체인을 위해 다중-계층 블록체인을 제안하였다. 다중-계층 블록체인에서는 비공개 블록을 암호화하여 저장한다. 이때, 대표노드의 가입과 탈퇴가 일어나는 DPoS 블록체인을 위한 비공개 암호 블록을 위한 암호화 요구사항을 정리하였다. 이때 중요한 점은 암호화 조각의 보호라기 보다는 신뢰할 수 없는 각 노드간의 공평성이 더 중요함을 설명하였다. 본 논문은 기존의 Hash-ElGamal 기법을 발전시킨 AES-ElGamal 기법을 이용하여 암•복호화하였으며, 비밀 조각 배분은 특정 대표노드가 온전한 비밀 값이나 다른 대표노드의 비밀 조각을 알지 못하게 Dealer없는 t-of-n Threshold 암호화 기법을 사용하였다. 또한, 대표노드가 가입과 탈퇴시 효과적으로 비밀값을 재분산하는 기법을 제안하였다. 마지막으로 제안 기법의 공평성과 성능을 분석하였다.

참고문헌

[1] Yaga, Dylan, et al., "Blockchain technology overview." arXiv preprint arXiv:1906.11078 (2019).

[2] 병적 별도관리대상자 명단 https://www.privacy.go.kr/wcp/pif/sch/personalInfoFileViewPopup.do?prsnInfoFileId=PIF_00000000071 2041

[3] Zheng, Zibin, et al., "An overview of blockchain technology: Architecture, consensus, and future trends." 2017 IEEE International Congress on Big Data (BigData Congress). IEEE, 2017.

[4] Hong, Jeongdae, et al., "Fair threshold decryption with semi-trusted third parties." Australasian Conference on Information Security and Privacy. Springer, Berlin, Heidelberg, 2009.

[5] Pedersen, Torben Pryds, "A threshold cryptosystem without a trusted party." Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1991.

[6] Daemen, Joan, and Vincent Rijmen, The design of Rijndael: AES-the advanced encryption standard. Springer Science & Business Media, 2013.

[7] ElGamal, Taher, "A public key cryptosystem and a signature scheme based on discrete logarithms." IEEE transactions on information theory 31.4 (1985): 469-472.

[8] Shamir, Adi. "How to share a secret." Communications of the ACM 22.11 (1979): 612-613.

[9] Reiter, Michael K., and Kenneth P. Birman, "How to securely replicate services." ACM Transactions on Programming Languages and Systems (TOPLAS) 16.3 (1994): 986-1009.

[10] Gennaro, Rosario, et al., "Robust threshold DSS signatures." International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1996.

[11] Frankel, Yair, Peter Gemmell, and Moti Yung, "Witness-based cryptographic program checking and robust function sharing." STOC. Vol. 96. No. 47. 1996.

[12] Gennaro, Rosario, et al., "Robust and efficient sharing of RSA functions." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1996.

[13] Pedersen, Torben Pryds, "A threshold cryptosystem without a trusted party." Workshop on the Theory and Application of of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1991.

[14] Boneh, Dan, and Matthew Franklin, "Efficient generation of shared RSA keys." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1997.

[15] Ostrovsky, Rafail, and Moti Yung, "How to withstand mobile virus attacks." PODC. Vol. 91. 1991.

[16] Herzberg, Amir, et al., "Proactive secret sharing or: How to cope with perpetual leakage." Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1995.

[17] Desmedt, Yvo, and Sushil Jajodia, "Redistributing secret shares to new access structures and its applications", Vol. 148. Technical Report ISSE TR-97-01, George Mason University, 1997.

[18] Frankel, Yair, et al., "Optimal-resilience proactive public-key cryptosystems." Proceedings 38th Annual Symposium on Foundations of Computer Science. IEEE, 1997.

[19] C. Gehrmann and Y. Desmedt, 'Truly Anonymous secret sharing', Manuscript.

[20] Li, Chuan-Ming, Tzonelih Hwang, and Narn-Yih Lee, "Threshold-multisignature schemes where suspected forgery implies traceability of adversarial shareholders." Workshop on the Theory and Application of Cryptographic Techniques. Springer, Berlin, Heidelberg, 1994.

[21] Desmedt, Yvo, Giovanni Di Crescenzo, and Mike Burmester, "Multiplicative non-abelian sharing schemes and their application to threshold cryptography." International Conference on the Theory and Application of Cryptology. Springer, Berlin, Heidelberg, 1994.

[22] Alon, Noga, Zvi Galil, and Moti Yung, "Efficient dynamic-resharing "verifiable secret sharing" against mobile adversary." European Symposium on Algorithms. Springer, Berlin, Heidelberg, 1995.

[23] 병역사항신고 및 공개 파일
https://www.privacy.go.kr/wcp/pif/sch/personalInfoFileViewPopup.do?prsnInfoFileId=PIF_00000000076460.

〔저자소개〕

정승욱 (Seung Wook Jung)
2005년 12월 : University of Siegen, 전자정보공학박사
현재 : 건양대학교 사이버보안공학과 교수
관심분야 : 개인정보보호, 블록체인, 네트워크 보안, 암호학 등
email :swjung@konyang.ac.kr