Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.2, Apr. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.2.139

익명인증서 및 블록체인 암호화로 익명성이 강화된 디지털화폐 모델*

윤 재 호,[1†] 김 용 민[2‡]
[1,2]전남대학교 대학원 정보보안협동과정 (대학원생, 교수)

CBDC Model with Enhanced Anonymity Using ID Certificate and Blockchain Encryption*

Jae-ho Yoon,[1†] Yong-min Kim[2‡]
[1,2]Interdisciplinary Program of Information Security,
Chonnam National University (Graduate student, Professor)

요 약

CBDC는 전자적 로그(log)에 의해 모든 기록이 남겨지는 전자지급수단과 유사한 특징을 가지고 있어 현금이 가지는 익명성을 충족하기는 어렵다. 이에 본 연구에서는 익명성 강화를 위해 디피헬만 키공유 알고리즘으로 거래내용을 모두 암호화하는 디지털화폐 거래모델을 제시하였다. 제시된 모델은 거래의 암호화를 통해 비연결성, 추적불가성 등의 익명성을 제공하고 있다. 또한 실명 인증이 되지만, 실제로는 익명을 이용하는 CBDC 인증서를 이용하였는데, 이 인증서를 통해 불법 등 추적이 필요한 거래는 권한이 부여된 기관 등에 의해 사후 추적도 가능하다.

ABSTRACT

CBDC has characteristics similar to e-payments in which all records are kept by logs, so it is difficult to satisfy the anonymity level of cash. Therefore, in this study, the CBDC model that encrypts all transaction contents using the Diffie-Hellman key sharing algorithm was presented to enhance anonymity. The proposed model provides unlinkability and untraceability. In addition, a CBDC certificate that uses pseudonym is used. Through this certificate, illegal transactions that require tracking can be tracked later by authorized institutions.

Keywords: CBDC, Blockchain, anonymity, anonymous certificate

I. 서 론

전세계적으로 사회 각 분야의 디지털전환이 가속화되는 가운데 금융환경도 빠르게 변화하고 있다. 특히 2009년부터 시작된 비트코인 등 암호화폐의 출현과 블록체인 기술의 확산은 각 국의 중앙은행들이 발행하는 현금과 더불어 디지털화폐(CBDC, Central Bank Digital Currency)의 연구·개발을 촉진시켰다. 중국, 우크라이나, 우루과이 등 3개국의 경우 이미 CBDC 시범사업을 추진 중에 있으며, 우리나라, EU를 비롯한 6개국은 CBDC 모의실험을 진행 중이다[1][2].

그러나, 중앙은행들이 검토하고 있는 대부분의 CBDC 발행모델은 기존 비트코인의 UTXO (Unspent Transaction Outputs) 모델을 차용하고 있어 현금의 가장 큰 장점인 익명성의 유지에

Received(12. 02. 2022), Modified(1st: 01. 10. 2023, 2nd: 02. 17. 2023), Accepted(02. 17. 2023)
* 이 논문은 2023년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(IITP-2022-0-01203)
† 주저자, 251na2026@jnu.ac.kr
‡ 교신저자, ymkim@jnu.ac.kr(Corresponding author)

어려움이 있다. 기본적으로 UTXO 기반의 모든 거래는 추적이 가능하여 거래 상대방의 지갑주소를 알 경우 거래 관련 자료를 대부분 수집할 수 있다. 익명성은 일반인들이 개인정보를 노출하지 않고 화폐를 사용할 수 있게 해주는 가장 큰 요인 중 하나이다. 전자지급수단이 확산되는 상황에서도 현금수요가 지속되는 이유 중 하나는 전자지급수단의 익명성 수준이 현금의 익명성 수준에 미치지 못하기 때문인 것으로 분석된다[3]. 또한, 익명 CBDC의 발행은 현금 흐름 정보가 은행 및 결제사업자들에게 공유되지 않아 CBDC로 거래하는 온라인 판매자에게도 정보 우위의 장점을 제공할 수도 있다[4].

이에 본 연구에서는 현금 수준의 익명성을 유지하면서도 필요에 따라 추적이 가능한 CBDC 모델을 제시하고자 한다.

II. 블록체인과 익명성

블록체인은 어떤 사람의 주소가 노출되면 해당 주소의 거래내역 및 암호화폐 흐름을 누구나 모니터링 할 수 있기 때문에 특정 주소의 모든 거래내역을 파악할 수 있게 되었다. 비트코인의 경우 지갑주소(공개키)와 이에 대응되는 실명을 연결할 수 없도록 하지만, 앞서 설명한 거래내역 추적불가 등 엄밀한 의미의 익명성 제공에는 한계가 있다. 즉 익명성 요건을 충족하기 위해서는 실명과의 비연결성(unlinkability)과 거래의 추적불가성(untraceability)이 필요하다. 이와 관련된 연구에서는 비연결성과 추적불가성을 다음과 같이 정의하고 있다[5].

- 비연결성 : 어떠한 2개 이상의 거래도 같은 사람이 송부했다는 것을 알 수 없어야 한다.
- 추적불가성 : 어떠한 거래와 관련하여 해당 입력에 의해 지급되는 출력은 다른 출력들 사이에서 익명이어야 한다.

비연결성과 추적불가성을 보장하기 위한 가장 보편적인 방법은 암호기술을 적용하여 블록 내에 정보를 숨기는 것이지만 이 경우 블록 내의 정보 확인이 불가능하여 이중지불 등의 부정거래 차단이 어려워진다. 이러한 익명성 문제를 해결하기 위해 지급·수취인 및 거래내역 등을 숨길 수 있는 Zcash, 모네로 등의 암호화폐들이 등장하였는데, 이러한 암호화폐들의 블록체인은 비연결성 및 추적불가성 관련 문제를 어느 정도 해소하면서 이중지불 등의 부정거래를 차단하는데 효과적이다. 다만, 이러한 기술들을 적용한 CBDC 모의실험 결과는 아직 발표되고 있지 않는데, 이는 익명성을 강화하는 암호기술로 인해 불법 등이 의심되는 거래의 추적이 어렵고, 해당 범죄자의 식별도 어렵기 때문이다. 이에 따라 비연결성과 추적불가성을 보장하면서 필요시 실명확인과 거래내역 추적이 가능한 블록체인에 대한 연구가 요구된다.

2.1 익명기술 동향

익명성과 프라이버시 보호를 위한 블록체인 기술과 관련하여 기존 플랫폼에 대해 Table 1.과 같은 기술들이 이용되고 있다[6].

Dash의 PrivateSend는 분산형 믹스가 일부 적용된 기법으로 믹스(Mix)는 수취인 주소 등을 무작위로 섞어 거래의 추적을 어렵게 하는 방법이다. 중앙믹스 및 분산형 믹스 기법에 대해서도 다양한 연구가 진행되고 있다[7][8].

한편, Zcash는 각 코인에 일련번호를 부여하고 코인이 사용된 후 장부에 해당 값을 공개하는 방식[9]인데, 이와 관련하여 공격자가 일련번호가 같은 2개의 지불메시지를 보낸 후 하나의 거래를 무효화하고 환불을 요구할 수 있는 취약점이 있다[10]. Zcash는 프라이버시 보호를 위해 zk-SNARKs (Zero-Knowledge Succinct Non- Interactive Argument of Knowledges)를 이용 중인데, 이는 모든 거래데이터를 암호화하여도 영지식 증명을 통해 거래의 적합성을 증명하는 기법으로 동형암호(Homomorphic encryption)의 한 종류로 볼 수 있다. 다만, zk-SNARKs는 파라메터 값 설정이 복잡하고 여러 단계의 수식 변경이 필요하기 때문에 연산속도가 비교적 느린 것으로 분석된다. 한편, 모네로 플랫폼은 링서명을 통해 송신자를 숨기고 스텔스 주소를 통해 수신자를 숨기는 기법을 이용하고 있는데, 링서명의 경우 송신자를 숨기기 때문에 이중지불에 취약할 수 있고 스텔스 주소의 경우 일회용 주소

Table 1. Anonymity methods

Platforms	Privacy	Anonymity
Dash	PrivateSend	
Zcash	zk-SNARKs	Stealth Address
Monero	RingCT	RingSign, Stealth Address

이기 때문에 안전성은 어느 정도 확보되지만, 거래마다 생성되는 주소이기 때문에 관리에 어려움이 있는 것으로 평가된다.

2.2 암호화 및 익명인증서

익명성 유지를 위해 다양한 암호기법이 개발되고 있으며, 비밀유지가 필요한 통신 및 메시지들은 암호로 변환·전달되고 이러한 암호는 수학적으로 풀기 어려운 문제에 기반하고 있다. 암호에 사용되는 대표적인 수학문제 중 하나는 이산대수 문제로 디피헬만(Diffie-Hellman) 키교환 방법[11]이나 엘가말(Elgamal) 암호시스템[12] 등이 이를 이용하고 있다[13].

한편 공개키 소유자의 정당성 확인을 위해 디지털 인증서를 이용하며, 기본적으로 인터넷과 같이 안전성이 보장되지 않은 공중망에서 사용자들이 상대방을 안전하게 인증하고 이를 통해 암호화된 데이터나 자금을 은밀하게 교환할 수 있게 해주는 것이 디지털 인증서의 역할이다. 이에 따라 디지털 인증서는 상대방 인증을 위해 현재 금융권에서 이용되고 있는 공동인증서와 같이 대부분 실명확인을 거쳐 실명으로 발급된다. 블록체인 참가자들의 정당성을 확인하기 위한 방법으로 디지털 인증서 기술이 매우 유용하지만 이를 바로 이용할 경우 실명노출의 위험이 있다. 이로 인해 익명으로 처리한 디지털 인증서의 이용을 고려할 필요가 있다.

익명인증서는 인증서 보유자의 실명이 인증서에 명시되지 않지만 필요시 정당 권한을 가진 기관 혹은 사람이 가입자의 신원을 확인할 수 있다. 이러한 점이 디지털 화폐 발행시 익명인증서의 도입이 필요한 이유이다. 또한, 디지털 인증서의 형태는 일련번호, 발행인, 정당성 검증 정보 등을 포함하고 있다는 점에서 지폐와 유사한 형태를 가지고 있다. 이러한 특성으로 인해 Furche 등의 연구에서는 인증서 기반의 CBDC 발행방식이 제안되었다[14]. 이 연구는 은행계좌를 기반으로 고객이 은행에서 자금을 이체할 때 은행간 거래를 중앙은행이 발행한 인증서 형태의 CBDC를 이용하는 방식이다. 이 방식은 은행계좌를 기반으로 발생하는 계좌이체 등에서 은행간 주고 받는 거래에 인증서 기반의 CBDC를 이용하기 때문에 개인 사이의 거래에 이용하기는 어렵다는 한계를 가지고 있다.

III. 익명성이 강화된 디지털화폐 모델

본 연구에서 제안하는 익명성이 강화된 디지털화폐 발행모델은 익명성 강화를 위해 2가지 주요 기법을 적용하였다. 우선 거래 당사자 인증은 디지털 인증서를 이용하지만 실명을 숨기기 위해 익명인증서를 적용하였다. 또한 거래 내역을 숨기기 위해 블록체인 내 거래정보를 디피헬만 키공유 알고리즘을 이용하여 암호화하였다. Fig. 1.은 제안하는 디지털 화폐 모델의 개념도이다.

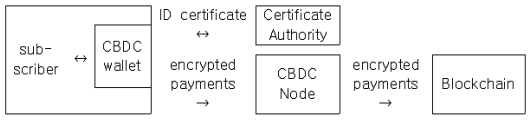

Fig. 1. Concept of proposed model

3.1 가입자 CBDC 지갑

디지털화폐 가입자는 CBDC 거래를 위해 우선 자신의 기기에 CBDC 지갑 프로그램(Wallet)을 설치할 필요가 있다. 가입자는 자신의 지갑을 통해 디지털 인증서의 발급, 잔액, 거래내역 등을 확인할 수 있다. 지갑프로그램은 IC칩 등 TPM(Trusted Platform Module) 형태로 외부 해킹공격 등에도 안전하게 보호되어야 하는 것이 원칙이다. 다만, 필요시 웹기반 또는 PC기반의 지갑도 안전성이 확보된다면 이용이 가능하다. Fig. 2.는 CBDC 지갑의 개념을 표현한다.

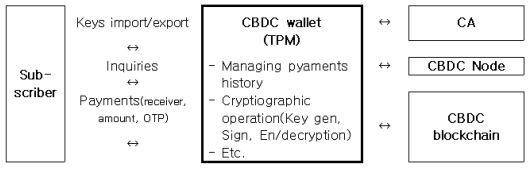

Fig. 2. CBDC wallet

3.2 CBDC 인증서 발급

일반적인 인증서의 발급은 등록기관(Registration Authority)에서 고객의 실명이 확인되면 인증기관(Certificate Authority)이 고객의 실명과 고객이 생성한 공개키를 넣어 인증서를 발행한다. 인증서 내의 공개키는 고객이 수행한 전자서명을 검증

하는 용도로 이용된다. 제안모델에서는 디지털화폐 가입자는 우선 통신사, 카드사 등 현재 제공되는 실명확인 서비스를 통해 실명을 인증한 후 실명 대신 표기될 ID(ex. 영문 4자리 + 숫자 6자리)와 자체 생성한 공개키를 인증기관에 송부한다. 또한, 향후 블록내 정보를 암호화하기 위한 유도값들을 인증서에 포함시키기 위해 관련 암호키 값도 같이 송부한다. Table 2.는 기존 인증서와 CBDC 인증서의 차이점을 설명하였다.

Table 2. Differences of X.509 and CBDC Cert.

Values	X.509 Cert.	CBDC Cert.
Subscriber Name	Real name	ID (pseudonym)
Public Key (for sign)	○	○
Encryption Key	X	○

본 연구에서는 인증서 발급과 관련된 CMP (Certificate Management Protocol, RFC4210) 과정[15]의 설명은 생략하도록 하며, 인증서 발급요청을 위한 주요 절차는 Fig. 3.과 같다.

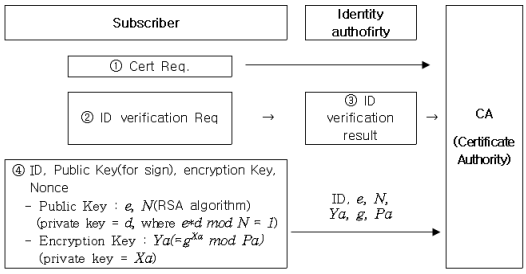

Fig. 3. Procedure of Certificate Request

단계 1.~3. 가입자가 인증기관에 접속하여 인증서 발급을 요청하면 인증기관은 실명확인기관으로 가입자를 안내하고 실명확인 절차를 거친다. 실명확인 결과는 인증기관에 송부된다.

단계 4. 실명이 확인된 가입자는 인증서 발급에 필요한 값들을 생성하여 인증기관에 송부한다. 인증서용 개인키 및 공개키는 인증기관의 선택에 따라 다양한 암호알고리즘을 이용할 수 있지만, 암호화용 키는 이산대수 문제에 기반한 암호알고리즘을 통해 공개키와 개인키를 선택해야 하며, 소수 Pa의 크기는 안전성을 위해 충분히 큰 크기의 수를 선택한다[16]. 인증서 발급을 위해 가입자가 생성하는 값은 Table 3.과 같다. 이중 인증서 발급을 위해 가입자는 ID와 더불어 e, N, g, P_a, Y_a 값을 인증기관에 송부한다.

Table 3. Crypto factors of subscriber

Crypto factors	Algorithm	Usage
e(private key), d, N	RSA	Signing and verification
Xa(private key), g, P_a, Y_a $(=g^{Xa} \bmod P_a)$	discrete logarithm	Payement encryption and decryption

단계 5. 한편, 인증기관은 가입자의 암호용 공개키(Y_a)와 가입자의 암호인자(g, P_a)를 CBDC 관리기관에 송부하면 CBDC 관리기관은 수신된 값에 자신의 개인키(C)를 연산한 값($g^{XaC} \bmod P_a$, $g^{C} \bmod P_a$)을 인증기관에 회신한다. 인증기관은 자신의 개인키(R)을 이용하여 암호 관련 키값(KC, ZC)을 생성한다. 인증기관과 CBDC 관리기관의 개인키인 R과 C값은 가입자의 P_a 값보다 작은 비트(bit)의 값을 선택한다. Fig. 4.는 인증기관과 CBDC 관리기관의 암호키값 연산과정이다.

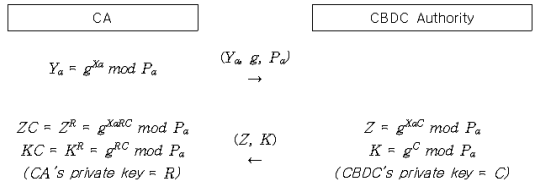

Fig. 4. Calculations of crypto key

단계 6. 인증기관은 가입자가 송부한 값(ID, 공개키(e, N), 암호키(Y_a, g, P_a) 및 암호키값(KC, ZC)이 포함된 인증서를 발급하고, 인증서 저장소에도 게시한다. 가입자는 두 개의 키(KC, ZC)값이

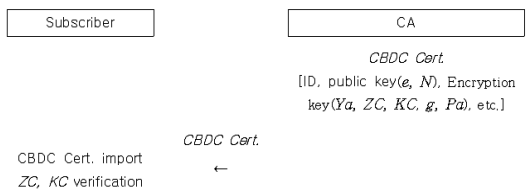

Fig. 5. CBDC Certificate Issuing

올바르게 포함되었는지 확인을 하기 위해 인증서 내의 KC를 개인키로 연산(=$(KC)^{Xa}$ mod P_a)하고 결과값이 ZC 값과 일치하는지 확인한다. Fig. 5.는 인증서를 발급받는 과정이다.

인증서의 형식은 RFC5280 형식[17]을 준용하며, ID는 가입자(subject) 필드, 암호화용 키는 Table 4.와 같이 확장필드(extensions)에 입력된다.

Table 4. Subject and extention fields

Subject Name	# CN=ID
extensions ::= SEQUENCE { extnID OBJECT IDENTIFIER, critical BOOLEAN DEFAULT FALSE, extnValue OCTET STRING }	 # OID TBD # set to "critical"

*extnValue*는 구분자('||')를 포함하여 다음의 값들을 순차적으로 표기한다.

· *extnValue* = P_a||g||Y_a(가입자 공개키: 암호용)||KC(기관키)|| ZC(암호키 유도값)

3.3 CBDC 거래 절차

본 연구에서 제안하는 CBDC 거래는 기본 거래모델과 기관 검증모델로 구분하며, 기본 거래모델은 앞서 기술된 CBDC 인증서를 이용하여 모든 거래정보를 암호화하며, 거래검증은 복호화가 가능한 수취인이 하는 모델이다. 기관 검증모델은 대체로 기본 거래모델과 유사하나 CBDC 관리기관이 거래를 검증한다는 특징이 있다.

3.3.1 기본 거래모델

거래의 세부절차를 살펴보기 위해 총 100원(70원+30원)을 보유하고 있는 가입자 A가 가입자 B에게 60원을 송금(40원 잔액)할 경우를 가정하였다.

단계 1. (수취인 인증서 획득) 가입자(A)는 인증서 저장소 또는 수취인(B)으로부터 수취인의 인증서를 획득한다.

단계 2. (A의 암호 관련키 생성) A는 예비 거래번호(Tx')값을 생성하고 거래내역(m)을 작성한 뒤 B의 인증서로부터 암호화에 필요한 키값들을 유도한다. 한편, A의 개인키는 매 거래마다 다른 값을 생성하기 위해 난수값과 같이 연산된다(A의 암호화 개인키 = Xa + 난수). 다만, 첫 거래의 난수값은 0이며 CBDC 지갑은 각 거래를 저장할 때 해당 난수값을 같이 저장하여 관리한다. 또한 예비거래번호(Tx')는 TA(Transaction dedicated Authenticator)[18]값이 이용되는데, 이 값은 기존 OTP값에 수취인의 ID를 합성하여 만들어진 값이다. 개인키 및 TA값은 Table 5.와 같다.

Table 5. Pre-transaction No. and Private key

A's factors	Values	Details
Pre-transaction No. *(Tx')*	TA(ex. AB367F01)	TA = Htc(OTP + ID_b Number part) · Htc is the first 8 numbers of Hexadecimal of hash result
Private key for encryption	X_a + Random	the first Random value is 0

단계 3. (메시지 전송) A는 관련 키 값들과 송금메시지를 Fig. 6.과 같이 CBDC Node에 전송한다. 노드, 기관 및 가입자들은 암호화된 채널을 이용해야 하며, 송신자의 서명값(S_a)이 첨부된다. 주요 값들의 유도과정은 Table 6.과 같다.

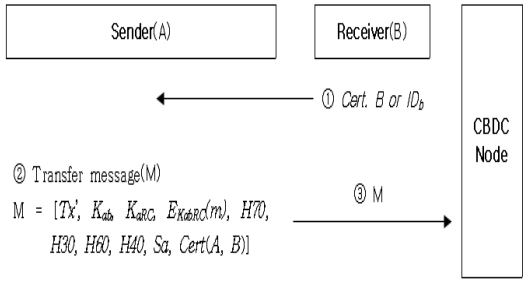

Fig. 6. Transfer message

단계 4. (노드의 메시지 검증) CBDC 노드(Node 1)는 우선 TA 값을 검증하고 A의 인증서를 통해 송금메시지의 서명값을 검증한 뒤 사용된 토큰값(H70, H30)이 유효한지 확인한다. 노드들은 사용 가능한(현재 유효한) 토큰 값들을 저장하고 있기 때문에 유효성 확인이 가능하다.

한편, 노드는 거래내역을 확인하기 위해 수취인 B에게 송금메시지 중에서 일부를 전달(KaRC,

Table 6. Crypto factors of A in transfer message

A's Crypto Factors	Initial vectors	Details
Sharing key(K_{ab})	encryption key(Y_b) from Cert. B	$(Y_b)^{Xa}=(g^{Xb} \bmod P_b)^{Xa} = g^{XaXb} \bmod P_b$
Receiver key(K_{aRC})	encryption key(KC) from Cert. B	$(KC)^{Xa}=(g^{RC} \bmod P_b)^{Xa} = g^{XaRC} \bmod P_b$
Payment message(m)	Spent:Tx2'||IDa||₩70, Spent:Tx3'||IDa||₩30, New:Tx'||IDb||₩60, New :Tx'||IDa||₩40 ('||' delimiter)	
Encrypted message ($E_{Kabrc}(m)$)	encryption key(ZC) from Cert. B	$K_{abRC} = (ZC)^{Xa} = (g^{XbRC} \bmod P_b)^{Xa} \bmod P_b$ $= g^{XaXbRC} \bmod P_b$
Token values	H70, H30, H60, H40	Hash values with prefix S, N(S=spent, N=New) H70 = S||H(Tx2'||IDa||₩70), H30 = S||H(Tx3'||IDa||₩30), H60 = N||H(Tx'||IDb||₩60), H40 = N||H(Tx'||IDa||₩40)
Singing value(Sa)	A's Singing Value of Transfer message : Tx', Kab, KaRC, E_{KabRC}(m), H70, H30, H60, H40	

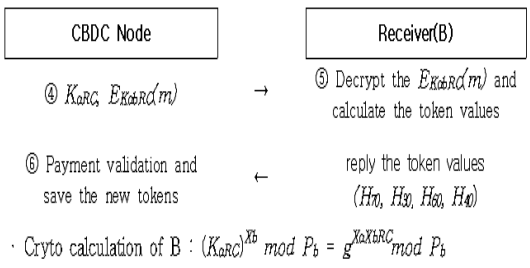

Fig. 7. Payment validation of CBDC node

EKabRC(m))한다. Fig. 7.은 메시지 검증과정을 나타낸다.

단계 5. (수취인 B의 거래확인) 가입자 B는 송금 메시지의 수신키값(K_{aRC})으로 부터 암호키값(K_{abRC})을 계산한 후에 암호화된 거래내역($E_{Kabrc}(m)$)을 복호화한다. 복호화된 거래내역(m)을 통해 수취인은 사용 토큰값(H70, H30) 및 신규 토큰값(H60, H40)를 계산하고 자신의 서명값을 첨부하여 노드에게 회신한다.

단계 6. (노드의 거래검증 및 신규토큰 저장) Node 1은 수취인 B로부터 전달받은 토큰값(H70, H30, H60, H40)을 확인한 후 신규 생성된 토큰값(H60, H40)을 저장한다.

단계 7. (노드의 전파값 생성 및 전파) Node 1은 최대 10,000개 또는 5분간 수집된 거래를 블록으로 만들고 합의를 위해 이 블록을 각 노드들에게 배포한다. 5분 동안 수집된 거래가 없을 경우 다음 5분간 거래가 수집될 때까지 대기한다. 한편 Node 1은 블록이 생성되면 거래번호(TX)를 A와 B에게 전송한다. 거래번호는 블록번호에 현재 거래번호 및 기존 예비 거래번호(TX)로 구성된다(TX = 블록번호+순번+TX'). 합의를 위해 전파되는 메시지의 구조는 Table 7.과 같다.

Table 7. Broadcasting message for consensus

Message 1. CBDC Block	
Block number	→ node 2
Header : Node name, timestamp, Number of transactions, Nonce, Hash value of previous block with Nonce	
Transaction details(n)	→ node 3
1:Tx', K_{ab}, K_{aRC}, E_{KabRC}(m1), H60, H40	
2:Tx2', K_{xx}, K_{xRC}, E_{KxxRC}(m2), Hx, Hy	→ node 4
…	
n:Txn', K_{xn}, K_{xnRC}, EK_{xnRC}(mn), Hn, Hm	…
Node's sign value	
Message 2. Token used	
- H70, H30,..., Hm(bit ascending order)	

첫 번째 메시지인 CBDC 블록에서 헤더에는 블록체인 구성을 위한 별도의 *Nonce* 값을 포함하는데, 이전 블록에 *Nonce* 값을 포함하여 해쉬한 결과

값도 같이 명시된다. 해쉬한 결과값에 일정 난이도를 부여하는 방법(ex. '00'로 시작)으로 비트코인의 작업증명처럼 블록의 안전성을 높일 수 있다. 두 번째 메시지는 각 거래에 사용된 토큰값이 표시된다.

단계 8. (노드합의 및 신규블록 생성) CBDC 블록값을 수신한 노드들은 우선 서명값을 검증하고 사용된 토큰값을 확인한 뒤 신규 토큰값을 저장한다. 합의는 작업증명(PoW)보다는 PBFT(PBFT, Practical Byzantine Fault Tolerance) 또는 Raft[19][20] 등 비교적 효율적인 알고리즘이 이용된다. Raft 알고리즘이 이용될 경우 고정된 노드에 거래가 집중되는 것을 방지하기 위해 주 노드가 일정 시간단위로 변경될 수 있도록 합의 알고리즘의 변경이 필요하다. 합의과정이 끝나면 노드들은 수신된 메시지 중 첫 번째 메시지인 CBDC 블록을 새로운 블록으로 추가·저장하여 블록체인을 만든다. 블록체인 정보는 Table 8.과 같으며, CBDC 노드는 CBDC 관리기관이 선택한 신뢰하는 기관에서 각각 운영하고, CBDC 관리기관도 블록체인의 모니터링을 위해 별도의 노드를 운영할 수 있다.

Table 8. Information of Block

Block number
Header : Node name, timestamp, Number of transactions, Nonce, Hash value of previous block with Nonce
Transaction details(n)
1:Tx', K_{ab}, K_{aRC}, E_{KabRC}(m1), H60, H40
2:Tx2', K_{xx}, K_{xRC}, E_{KxxRC}(m2), Hx, Hy
…
n:Txn', K_{xn}, K_{xnRC}, EK_{xnRC}(mn), Hn, Hm
Node's sign value of Block

3.3.2 기관 검증모델

앞서 설명한 기본 거래모델의 경우 거래 검증을 위해 수취인은 항상 온라인이어야 한다는 제약이 있다. 이는 거래내역을 복호화할 수 있는 사람이 수취인이기 때문인데, 거래내역은 공유키값(Kab)를 통해서도 복호화가 가능하다. 이러한 점을 이용하여 거래 검증을 CBDC 관리기관과 인증기관의 협조로 수행하는 모델을 고려할 수 있다. 기본 모델의 경우 노드의 메시지 검증을 위해 수취인 B의 메시지 송·수신 절차(거래절차 ⑤)가 필요했는데, 기관 검증모델

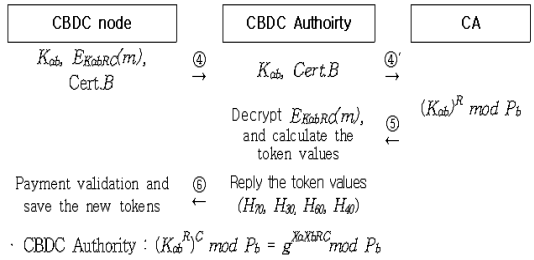

Fig. 8. Concept of Authority Validation Model

의 경우 수취인이 아닌 CBDC 관리기관이 메시지 검증값을 회신한다. Fig. 8.은 기관 검증모델의 개념도이다.

한편, 기관 검증모델에서 각 토큰값에 아래와 같이 금액을 표시하는 방안을 고려해 볼 수 있는데, 거래에 이용되는 모든 토큰값에 금액표기를 추가하는 방법이다.

기존 : H70 → 변경 : H70||₩70

이 경우 유효한 토큰의 전체 합이 총 통화량이 되므로 CBDC 관리기관은 블록내 정보들을 개별적으로 확인하지 않고도 토큰에 표시된 금액을 통해 전체 통화량을 확인할 수 있다. 또한, 해킹, 오류 등에 의해 정당하지 않은 토큰이 생성되었을 경우 총 통화량 값이 변하게 되는데, 통화의 변화를 통해 정당하지 않은 거래를 손쉽게 식별할 수 있다. 각 노드들도 신규 생성된 토큰의 합과 사용된 토큰의 합이 같다는 것을 통해 거래메시지를 추가로 검증할 수 있다. 다만, CBDC 블록에는 거래 메시지가 암호화되어 있고 사용된 토큰값도 확인할 수 없으므로 일반 가입자들이 파악할 수 있는 정보는 없다.

3.4 제안 모델의 분석 및 평가

3.4.1 시스템 프로토타입

본 연구에서 제안한 시스템의 프로토타입은 파이썬 프로그램 언어를 이용하여 구현하였다. 시스템 전체 모델의 구현이 아닌 프로토타입으로 키공유를 위한 기본적인 암호모듈과 블록체인 생성모듈을 중심으로 본 연구에서 제시한 기본 모델을 개발하였다. 구체적으로는 2개의 거래를 생성한 후에 각 거래마다 새로운 블록이 생성되는 구조로 원시블록(Genesis block)을 포함하여 총 3개의 블록을 생성하는 시스

Table 9. Example of transaction process

Payer(ID)	OTP	Holding Tokens	Payee(ID)	Transfer Amount
A: sang801039 (X_a = 111)	290315	₩70 + ₩30 (58a2b072||sang801039||70won) (ef71c099||sang801039||30won)	B: yoon262613 (X_b=456)	₩60
C: yshe872591 (X_c = 222)	123456	₩80 (b992b013||yshe872591||80won)	D: chan427010 (X_d=789)	₩30

step 1. Tx' A : 87edf398, Tx' C : 4fc15e38
step 2. A's Crypto factors : K_{ab}=18099, K_{aRC} = 7968, K_{abRC} = 12798
step 3. Encryption(AES128 CBC mode)(A's message)
 - E_{KabRC}(m) : gAABjEFmQ6PQGB9....3K9eGPiDwbYzY
step 4. Token values(examples)
 - spent H70(A) : 42db4a1e3f7b87a2079f12c316745d4509e7e4696f571546d28071bfaf81f52c
 - new H50(C) : b3234d220bfd6a247622504fb1955fbb830efea6ac40360194d0affbdd0ce5d4
step 5. Transfer message(M)(A's message)
  A: [Tx' : 87edf398, Kab : 18099, KaRC : 7968,
       EKabRC(m) : gAAAAABjEFmQ6PQGB9....3K9eGPiDwbYzY2QE4P4yh
       Spent : 42db4a1e3f7b87a2079f12c316745d4509e7e4696f571546d28071bfaf81f52c,...
       New : b3234d220bfd6a247622504fb1955fbb830efea6ac40360194d0affbdd0ce5d4
       S_A : 5ILx2NqPTDBj.....KW2J1thgxV7xwReFc], Acert.der, Bcert.der
step 6. Block information(example)
==========
Index : 1
Constructor(Node name) : Node1
Timestamp : 2022-09-08 09:02:05.049865
The number of transactions : 2
Nonce : 164
Previous_hash : 00f4aa6bd68c7238ec2773411a0afe0210d99939dcb2a753af1e6b12e54b60cb
Transactions :
1 : 'Tx': '87edf398', 'Kab': '18099', 'KaRC': '7968', 'EKabRC(m)': 'gAAAAABjEFmQ6PQGB9...3KGPiD
    wbYzY2QE4P4yh', 'New': '51c836d8f368fb8be5782d8fa33f7b7adfee61a995c644c24670c6bf8f70bdb8'
2 : .....
Signature : xv0ZRE8KW3R...q0Z2LPFEXphuqW8
==========

Table 10. Pseudo code of blockchain

```
# Python Code
def proof_of_work(zeros=2):  # difficulties
    global nonce
    global proof
    nonce = 0
    proof = generate_hash()
    while proof[:2] != '0'*zeros:
        nonce += 1
        proof = generate_hash()
    return proof, nonce

def generate_hash():  # Hash of block with nonce
    block_contents = str(index) + str(node_name) + str(timestamp)+ str(number_tr)+ str(nonce_pre)
                     + str(previous_hash) + str(transactions) + str(signature) + str(nonce)
    block_hash = sha256(block_contents.encode())
    return block_hash.hexdigest()

proof, nonce = proof_of_work() # for genesis block
for i in range(1,len(list)+1):
    timestamp = datetime.now()
    number_tr = len(list)
    nonce_pre = nonce
    previous_hash = proof
    transactions = list[i-1]
    proof_of_work()
```

Table 11. Comparison between proposed model and platforms

Categories		Bitcoin	Zcash	Propsed Model
Anonymity	unlinkability	○ (address:public key)	○ (address:public and stealth)	○ (block encryption)
	untracability	X	△*	○
Block generation		PoW	PoW(Equihash)	Consensus(Raft)
Payment Authentication		Sender's signature	Sender's signature	Sender's signature (ID Cert.) and TA
Wallet		Not fixed	Not fixed	based on TPM
Payment validation		UTXO	zk-snark	UTXO and receiver (or Authority)

* shielded(stealth) address used only

템 코드를 구현하였다.

본 연구는 작업증명이 아닌 Raft 합의 알고리즘을 전제로 블록체인이 생성되며, 개별블록에 작업증명을 적용하여 안전성을 증가시켰다. 프로토타입의 시나리오는 Table 9.의 정보를 기반으로 구성되었으며, Fig. 10.은 프로토타입 의사코드 개요이다.

3.4.2 안전성 및 효율성 분석

제안한 모델의 경우 개인키(X_a) 및 난수값에 의해 매 거래마다 각자의 암호키가 변경되므로 어떠한 2개 이상의 거래도 같은 사람이 송부했다는 것을 알 수 없어 비연결성을 만족한다. 또한, 블록에 저장된 암호화된 메시지 및 사용·생성 토큰값은 서로 연결되어 있지 않으므로 추적불가성도 충족한다. 또한, 디지털 인증서를 처리하는 노드들도 모든 거래가 익명 인증서에 포함된 ID를 기준으로 수행되므로 인증기관 및 실명 확인기관의 협조없이 가입자 신원확인은 불가능하다. 인증기관 및 CBDC 관리기관은 불법으로 의심되는 거래의 경우 관련 노드로부터 지급인·수취인의 인증서를 획득하고 두 기관 협의하에 공유키값(K_{ab})으로부터 암호키값(K_{abRC})을 유도한 뒤 거래 메시지를 복호화 할 수 있다. 이 경우에도 거래당사자의 ID만 확인되므로 고객 실명확인을 위해서는 인증기관의 실명확인 협조가 필요하다. 한편, 제안 모델과 현재 운영중인 플랫폼인 비트코인, zcash와 안전성 및 효율성을 Table 11.에 비교하였다.

앞서 기술한 바와 같이 제안모델은 자금이체 정보를 암호화하였기 때문에 일반 가입자 등에 공개된 블록체인을 통해 알 수 있는 거래 관련 정보는 없다. 또한, 가입자가 지급지시를 할 때 인증기관을 통해 발급받은 가명 인증서를 통해 정당성이 확인되고 추가로 TA 값을 통해 가입자를 인증하므로 거래의 안전성이 증가된다. 한편, 대부분의 플랫폼에서는 지갑과 관련하여 특별한 제약사항이 없지만 제안모델은 안전성이 우수한 TPM 기술을 기본적으로 적용하였다. 블록생성과 관련하여서도 작업증명이 아닌 효율성이 우수한 Raft 등의 합의알고리즘을 적용한다.

IV. 결 론

전자지급수단은 편의성을 제공해 왔으나, 전자적 특성으로 인해 익명성에 한계를 갖게 되었다. 현금의 보완 수단으로 CBDC가 논의되고 있지만, 전자지급수단과 유사한 특징으로 인해 CBDC가 현금이 가지는 익명성을 충족하기에는 부족한 것이 현실이다. 현재 익명성을 보장하는 기술로 영지식증명 등이 많은 주목을 받고 있으나, 영지식증명 방식을 중앙은행 디지털화폐에 적용하기에는 아직 많은 검증이 필요할 것으로 보인다. 이에 본 연구에서는 오랜기간 검증되고 널리 이용되는 디피헬만 암호알고리즘을 이용하여 익명성이 강화된 디지털화폐 모델을 제시하였다. 제시된 모델은 비연결성, 추적불가성 등 익명성을 제공하고 있으며, 이와 더불어 기존 인증서와 달리 익명을 이용하는 CBDC 인증서를 이용하였고, 익명인증서에 고유의 공개키뿐만 아니라 암호화를 위한 공개키 요소를 여러 개 포함하는 모델을 제안하였다. 이를 통해 당사자간 이용되던 디피헬만 알고리즘을 다자간 알고리즘으로 확장 적용하였다.

본 연구에서 제안된 모델은 이산대수 문제에 기반한 암호알고리즘으로 향후 효율성 개선을 위해 이산대수 문제를 타원곡선 암호알고리즘으로 변형하여 적

용하는 방안의 연구가 필요하며, 실제 운용환경을 고려한 효율성 검증에 대한 추가적인 연구도 필요하다. 한편, 제안된 모델은 부분 암호화가 필요한 다른 플랫폼으로의 변형도 가능하다. 예를 들어, 토큰의 소유주를 명시적으로 나타낼 필요가 있는 NFT (Non-Fungible Token) 플랫폼의 경우 금액이 아닌 ID의 표시가 중요한데 ID를 토큰에 첨부하고 다른 정보는 암호화하여 숨기는 방식의 NFT 플랫폼으로 변형할 수 있다. 또한 향후 오프라인에서 개별 토큰의 유효성을 확인할 필요가 있을 수 있는데, 이 경우 인증서의 실시간 확인 프로토콜인 OCSP(online certificate status protocol)와 유사한 프로토콜을 개발하여 각 노드들이 토큰의 유효성을 손쉽게 검증할 수 있는 서비스를 제공할 필요가 있다.

References

[1] Bank of Korea, Central Bank Digital Currency global discussion trends by key issues, Jan. 2022.

[2] Jae-ho Yoon and Yong-min Kim, "Comparative Analysis on Digital Currency Models and Electronic Payments," The Journal of the Korea Contents Association, 22(7), pp. 63-72, Jul. 2022.

[3] Bank of Korea, Survey results on payment methods and mobile financial service user behavior in 2021, May. 2022.

[4] Toni Ahnert, Peter Hoffmann and Cyril Monnet, "The digital economy, privacy, and CBDC," ECB working paper, May. 2022.

[5] A Kumar, C.Fisher, S.Tople, and P. Saxena, "A Traceability Analysis of Monero's Blockchain," National University of Singapore, Apr. 2017.

[6] Ji-sun Park and Sang-uk Shin, "Analysis of Blockchain Platforms from the Viewpoint of Privacy Protection," Korean Society For Internet Information, 20(6), Dec. 2019.

[7] Yun-ho Lee, "Mix-based Decentralized Anonymous Transaction for Blockchain," Korean Society For Internet Information, 21(6), Oct. 2020.

[8] T. Ruffing, P. Moreno Sanchez, and A Kate, "CoinShuffle:Practical Decentralized Coin Mixing for Bitcoin," LNCS vol. 8713, pp. 345-364, 2014.

[9] Ji-su Son, et. el, "Privacy-preserving On-chain Blockchain based on zk-SNARK," Computing Science and Engineering KCC2019, pp. 1978-1980, Jun. 2019.

[10] Ji-won Hur, et el., "Accountable Privacy for Decentralized Anonymous Payments based on zk-SNARK," Computing Science and Engineering KCC2019, pp. 1993-1995, Jun. 2019.

[11] Whitfield Diffie and Martin Hellman, "New Direction in Cryptography," IEEE Transactions on Information Theory, vol. 22, no. 6, Nov. 1976.

[12] Taher Elgamal, "A public key cryptosystem and a signature scheme based on discrete logarithms," IEEE transactions on Information Theory, vol. 31, no. 4, Jul. 1985.

[13] Su-hyun Oh, et el., "A study on the cryptographic scheme security based on discrete logarithm problems," Review of KIISC, 9(1), pp. 43-68, Mar. 1999.

[14] Andreas Furche and elvir Sojli, "Central bank issued digital cash," https://doi.org/10.2139/ssrn.3213028, Aug. 2018.

[15] C. Adams, S. Farrell, and T. Kause, T. Monoen, "Internet X.509 Public Key Infrastructure Certificate Management Protocol(CMP)," IETF RFC4210, Sep. 2005.

[16] Telecommunications Technology Association Standard, "Digital Signature Mechanism with Appendix-Part 2: Korean Certificate-based Digital Sign-

ature Algorithm(KCDSA)," TTAK.KO-12.0001/R4, Dec. 2016.
[17] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List(CRL) Profile," IETF RFC 5280, May. 2008.
[18] Jae-ho Yoon, "Authencation password generator and its financial system," Korea Patent No. 1020150044908, Mar. 2015.
[19] Diego Ongaro and John Ousterhout, "In search of an Understandable consensus Algorithm," Stanford University, Jun. 2014.
[20] Youn-a Min, "A Study on PBFT Consensus Process Considering Communication Cost Efficiency of Blockchain Network," Journal of Korean Institute of Information Technology, 18(4), pp. 101-107, Apr. 2020.

〈저자소개〉

윤 재 호 (Jae-ho Yoon) 정회원
1997년 2월: 인하대학교 전자공학과 (공학사)
2004년 8월: 세종대학교 소프트웨어공학과 (공학석사)
2022년 3월~현재: 전남대학교 정보보안협동과정 박사과정
〈관심분야〉 정보보호, 전자상거래 보안 등

김 용 민 (Yong-min Kim) 종신회원
2002년 8월: 전남대학교 전산통계학과 (이학박사)
2006년 3월~현재: 전남대학교 문화콘텐츠학부/정보보호협동과정 대학원 교수
〈관심분야〉 시스템 및 네트워크 보안, 전자상거래 보안, 융합보안 등