*Journal of The Korea Institute of Information Security & Cryptology* ISSN 1598-3986(Print)
VOL.31, NO.5, Oct. 2021 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2021.31.5.863

# 블록체인 기반 가상자산 관리를 위한 (1,3) 분산키의 비신뢰 기반 안전한 분산 복구 프로토콜*

배 경 일,[1†] 박 준 후,[2] 류 재 철[3‡]
[1]아톰릭스랩 (이사), [2]아이오트러스트 (연구원), [3]충남대학교 (교수)

# Secure Recovery Protocol of (1,3) Distributed Key Share with Trustless Setup for Asset Management in Blockchain*

Kyoungil Bae,[1†] Junhoo Park,[2] Jaecheol Ryou[3‡]
[1]Atomrigs Lab (Managing Partner), [2]IoTrust (Researcher),
[3]Chungnam National University (Professor)

## 요 약

비신뢰 기반 분산키 생성은 다수의 참여자가 개인키를 모르는 상태에서 개인키의 샤미르 비밀 공유를 공동 생성하는 프로토콜이다. 또한 이를 임계값 서명으로 확장할 경우 개인키를 복원하지 않고도 다수 참여자간의 암호 프로토콜을 통해서 디지털 서명을 생성할 수 있도록 한다. 본 연구는 활용성이 높은 (1,3) 샤미르 비밀 공유 구조에서 한 공유 값을 분실할 경우 동일한 개인키에 대한 전체 공유 값을 비신뢰 기반으로 재생성해서 공유 구조를 복구하는 프로토콜을 제안한다. 제안 프로토콜은 정확성과 기밀성 측면에서 분산키 생성과 동일한 보안 요건을 갖춘다. 블록체인 기반 가상자산 관리에 적용할 경우 안전한 개인키 관리와 서명 권한의 위임 및 불능화를 가능하게 한다.

## ABSTRACT

Distributed key generation (DKG) with trustless setup is a cryptographic protocol that distributes Shamir secret shares of a private key to participants while keeping the actual private key hidden to the participants. Also, by extending it to a threshold signature protocol, digital signatures can be generated without construction of private keys. This paper proposes a recovery protocol maintaining trustless setup assumptions, in particular to the useful (1,3) share structure. The proposed protocol meets same levels of security requirements with DKG in terms of correctness and secrecy. The protocol can also enable delegation and revocation of digital sign rights for blockchain-based asset management.

**Keywords:** Blockchain, Key Recovery, Distributed Key Generation, Secret Sharing

## I. 서 론

최근 공개 블록체인 기술을 기반으로 발행되는 가상 자산의 가치와 활용 범위가 확대되고 있다. 공개 블록체인에서는 중앙화된 은행이나 자산관리 기관이 존재하지 않기 때문에 가상 자산 소유자가 개인키로 직접 가상 자산을 관리하고 전송하게 된다. 따라서 개인키의 안전한 관리와 사용은 곧 가상 자산의 보안성과 직결된다.

소유자가 개인키를 분실할 경우 블록체인 상의 자

Received(09. 16. 2021), Modified(10. 08. 2021), Accepted(10. 09. 2021)
* 이 논문은 2020년도 정부(과학기술정보통신부)의 재원으로 정보통신기획평가원의 지원을 받아 수행된 연구임(No. 2020-0-00229, 블록체인 기반의 멀티 레벨 인증키 관리 및 복구 응용 플랫폼 개발)
† 주저자, k@atomrigs.io
‡ 교신저자, jcryou@cnu.ac.kr(Corresponding author)

산을 통제하는 것이 불가능하다. 그리고 제3자가 개인키를 해킹할 경우 제3자가 블록체인 자산을 마음대로 통제할 수 있게 된다. 공개 블록체인에서는 소유자가 개인키를 보관하고 사용하기 위해서 소프트웨어 혹은 하드웨어 전자지갑[1,2,3,4]을 이용하는데 대부분의 경우 전자지갑 로컬 저장소에 개인키를 저장한다. 이러한 방식으로 개인키를 보관할 경우 단일 지점에서 개인키가 분실 혹은 유출될 수 있는 단일 실패 지점(Single Point of Failure)이 존재하게 된다. 즉 제3자가 한 지점의 데이터나 디바이스를 탈취하는 것만으로 타인의 개인키를 확보하는 것이 가능하다.

이러한 단일 실패 지점 문제의 해결 방안은 암호학자들이 장기간 연구해 온 주제이다. 1980년대부터 새로운 프로토콜들이 제안되었으며 특히 최근 블록체인 분야에서의 활용성이 높아짐에 따라 활발하게 연구 개발이 진행되고 있다. Adi Shamir는 개인키를 가지고 있는 딜러가 개인키를 $n$명의 참여자에게 분배하고 이중 임의의 참여자 $t+1(t+1 < n)$명이 개인키를 복원할 수 있는 샤미르 비밀 공유(Shamir Secret Sharing)를 제안했다[5]. $t$명 이하의 참여자로는 개인키를 복원할 수 없기 때문에 $t$를 임계값(threshold)이라고 하고 이를 $(t,n)$ 샤미르 비밀 공유라 한다.

이후 1990년대에는 딜러가 없이 다수의 참여자 $n$명이 아무도 개인키를 모르는 상태에서 $(t,n)$ 샤미르 비밀 공유를 생성하는 안전한 DKG(Distributed Key Generation)프로토콜들이 제안되었다[6,7]. ECDSA(Elliptic Curve Digital Signature Algorithm)[8]를 사용하는 공개 블록체인에서의 (1,3) 샤미르 비밀 공유 사용 예를 들어보자. Alice, Bob, Carol이 DKG 프로토콜로 샤미르 비밀 공유를 생성하며 이때 각 참여자는 각자의 샤미르 비밀 공유 값 외에 이로부터 복구 가능한 개인키는 알지 못한다. 그러나 프로토콜 과정에서 개인키에 대한 공개키와 이로부터 파생되는 블록체인 주소를 알게 된다. 개인키는 세 명중 두 명의 샤미르 비밀 공유 값들을 이용해서 복원 가능하다.

그런데 이 경우 블록체인 가상 자산을 외부로 이체하기 위해서는 두 명이(예를 들면 Alice와 Bob) 샤미르 비밀 공유 값을 공개하고 개인키를 복원한 후 디지털 서명을 생성해야 한다. 바로 이 과정에서 또 다른 단일 실패 지점이 발생한다. 외부로부터의 해킹 위협도 있지만 참여자 둘 중 한 명이 개인키를 알게 되므로 아무도 단독으로 개인키를 생성하지 못하던 초기의 장점이 없어지게 된다.

이러한 문제를 해결하기 위해서 참여자들의 샤미르 비밀 공유 값을 공유하거나 개인키를 복원하지 않고도 디지털 서명을 생성할 수 있는 임계값 서명 프로토콜이 개발되어 왔다. Gennaro 등은 Secure MPC(Multiparty Computation)프로토콜[9]을 적용하여 ECDSA에 적용 가능한 효율적인 DKG 프로토콜과 서명 프로토콜을 제안하였으며[10], 유사한 DKG 프로토콜을 기반으로 서명 프로토콜을 개선하는 연구들이 지속되고 있다[11,12,13].

본 논문에서는 Gennaro의 DKG 프로토콜을 확장 적용하여 비밀 공유 값의 복구 프로토콜을 제안한다. 실제 어플리케이션에서는 사용자가 공유 값을 분실하는 것은 자주 발생할 수 있는 상황이다. 특히 (1,3) 샤미르 비밀 공유의 경우 한 참여자의 공유 값 분실 후 다른 참여자가 공유 값을 분실할 경우 개인키 복원이 불가능해 진다. 그러므로 한 참여자의 공유 값 분실 시 동일한 개인키를 복원할 수 있는 새로운 공유 값을 복구하는 프로토콜이 반드시 필요하다. 또한 복구과정에서 참여자 누구도 개인키를 알 수 없어야 한다.

위의 예를 이용해서 설명해 보자. Alice가 공유 값을 분실할 경우 Bob과 Carol의 공유 값들로 개인키를 복원하는 것은 가능하나 만약에 Bob이 이후 공유 값을 분실할 경우 개인키를 영원히 복원할 수 없게 된다. 또한 Alice의 공유 값을 탈취한 제3자 Eve가 있다고 가정해 보자. Eve가 Alice인 척하고 Bob 혹은 Carol과 함께 임계값 서명 프로토콜을 실행할 경우 디지털 서명을 생성할 수 있다.

따라서 Alice가 공유 값을 분실할 경우 Alice가 새로운 공유 값을 생성함과 동시에 Alice의 기존 공유 값을 불능화(revocation) 시키기 위해서 Bob과 Carol의 공유 값도 변경시켜야 한다. 이때 세 참여자가 새로 가지는 공유 값 중 두개로 기존의 개인키가 복원 가능해야 한다.

가장 쉬운 방법은 다른 참여자(Bob과 Carol)들의 공유 값들로 개인키를 복원한 후 이를 다시 샤미르 비밀 공유로 삼자에게 분배하는 것이다. 그러나 세 참여자중 한 명 혹은 제3자가 개인키를 알게 되고 따라서 단일 실패 지점이 발생하게 된다.

본 논문에서는 참여자들 혹은 제3자에게 개인키를 유출시키지 않으면서 기존 개인키를 복원할 수 있는 새로운 샤미르 공유 값들을 참여자들이 가지게 되는

안전한 복구 프로토콜을 제안한다. 각 참여자들은 개인키는 물론 다른 참여자들의 공유 값도 알지 못한다.

논문에서는 구조를 단순화하기 위해서 일반적인 $(t,n)$방식이 아닌 (1,3) 방식의 샤미르 비밀 공유를 가정하여 설명한다. (1,3) 방식은 가상 자산 관리, 커스터디(custody), 서명 위임 등의 다양한 어플리케이션에 적용 가능한 유용한 구조이다. 예를 들면 블록체인 가상자산 커스터디 업계을 대표하는 Bitgo의 경우 세 개의 분산키를 생성해서 사용자와 Bitgo가 두 개의 키로 공동 서명하고 나머지 하나의 키는 사용자 키의 분실을 대비해서 별도 보관하는 (1,3) 분산키 구조를 사용하고 있다[14].

그런데 Bitgo에서 비트코인 가상자산에 적용하는 기술 구조의 경우에는 사용자가 분산키를 분실할 경우 분산키를 복원하기 보다는 새로운 (1,3) 구조 분산키와 주소를 생성한 후 Bitgo가 보유한 분산키와 세 번째 분산키를 이용해서 기존 가상자산을 새로운 주소로 이체하는 방식이다. 즉 해당 주소의 분산키 복원이라기보다는 자산의 분실을 막은 비상수단의 의미라고 볼 수 있다.

개인키 복원을 위해 사용하는 또 다른 대표적인 방식은 클라우드 사업자가 제공하는 EaaS (Encryption as a Service)를 이용하는 것이다[15]. 대부분의 클라우드 사업자들이 HSM (Hardware Secure Module)을 이용해서 데이터의 암복호화 서비스를 제공하고 있다[16, 17]. 사용자는 EaaS를 이용해서 개인키를 암호화한 후 암호문을 자체 저장소 혹은 제삼의 클라우드에 백업한다. 이후 개인키를 분실할 경우 EaaS를 통해서 백업한 암호문을 개인키로 복원한다. 이 경우 클라우드 사업자에게 종속된다는 문제가 있으며 더욱이 개인키의 암호문을 분실할 경우에는 개인키를 복원할 수 없다는 문제가 발생한다. 본 논문에서는 특정 클라우드 사업자에게 종속되지 않고 일반적인 컴퓨팅 환경에서 적용 가능하며 (1,3) 구조에서 한 공유 값의 분실 시 블록체인 주소를 변경하지 않고도 새로운 샤미르 공유 값을 생성할 수 있는 프로토콜을 제안한다.

본 논문의 구성은 다음과 같다. 2장은 사전 지식으로 본 논문을 이해하는 데에 필요한 암호 기법을 설명하고, 3장은 관련 연구로서 샤미르 비밀 공유와 DKG 프로토콜을 소개한다. 4장은 본 논문의 복구 프로토콜을 제안하고, 5장에서 프로토콜의 보안 요구사항을 검증한다. 6장은 프로토콜의 계산 속도 측정 결과를, 7장은 적용 가능한 어플리케이션을 소개한다. 마지막으로 8장에서 논문의 내용을 요약하고 결론을 맺는다.

## II. 사전 지식

이 글에서의 모든 기법은 ECDSA를 기준으로 설명한다. 타원곡선에 대한 베이스 유한체는 소수 $p$에 대해 $\{0,1,2,...,p-1\}$인 $Z_p$이고, $Z_p$를 베이스로 하는 타원곡선 군, $E(Z_p)$는 소수 $q$를 차수로 가진다. $G$는 타원곡선 군의 생성자(generator)이다.

$(t,n)$ 샤미르 비밀 공유 생성 혹은 복구 프로토콜의 참여자는 $\{P_1,...,P_n\}$으로 표기하고 이들이 가지는 샤미르 비밀 공유 값은 각각 $\{x_1,...,x_n\}$으로 표기한다. 임의의 $t+1$ 참여자가 복원 가능한 개인키는 $sk$, 이의 공개키(즉 $sk\cdot G$)는 $pk$로 한다.

특정 집합 $A$에서 원소 $a$를 랜덤하게 추출할 경우에는 $a \xleftarrow{\$} A$로 표기한다.

### 2.1 주요 가정

첫 번째 가정은 ECDLP(Elliptic Curve Discrete Logarithm Problem)를 풀 확률은 무시할 만하다(negligible)는 것이다. 다시 말하면 프로토콜에서 사용되는 타원곡선 군의 차수 $q$가 충분히 클 경우 타원곡선 군에 속한 원소 $K$가 주어질 경우 $K=k\cdot G$를 만족하는 $k$를 찾는 것은 기술적으로 거의 불가능하다.

두 번째 가정은 프로토콜 진행 중에 특정 참여자가 프로토콜에서 정해진 형태로 수행하지 않는(dishonest) 것을 발견할 경우, 다른 참여자들은 프로토콜을 즉시 멈춘다(abort). 예를 들면 다른 참여자가 보낸 영지식증명의 검증이 실패할 경우 프로토콜을 중지한다. 따라서 일부 악의적 참여자가 존재해도 결과를 내는 완결성(robustness)은 지원하지 않는다.

세 번째 가정은 단일 참여자가 다수의 참여자에게 동일한 메시지를 보내는 공개 채널과 두 참여자 간의 비밀 정보 전송을 위한 비밀 채널이 존재한다는 것이다.

### 2.2 샤미르 비밀 공유

샤미르 비밀 공유는 비밀정보 $s$를 알고 있는 딜러

가 $s$를 $n$명에게 분배하고 $n$보다 작거나 같은 임의의 $t+1$명이 비밀정보 $s$를 복원할 수 있도록 한다. 딜러는 $s$를 Y-절편으로 하는 랜덤한 $t$차 다항식 $f(x)$를 생성한다.

$$f(x) = a_0 + a_1 x + a_2 x^2 + \cdots + a_t x^t \pmod q$$
$$where \;\; a_0 = s$$

이후 딜러는 참여자 $P_i$에게 $f(i) \pmod q$를 전달한다. 이중 임의의 $t+1$명이 비밀정보 복원에 참여하게 되면 Lagrange 보간법(interpolation)[18]을 이용해서 $t$차 다항식 $f(x)$를 복원하고, $f(0)$인 비밀 정보 $s$를 얻게 된다.

### 2.3 Lagrange 보간법

이 중 $t+1$명의 참여자가 개인키 복원에 참여할 경우 복원 참여자의 집합을 $\mathbb{C}$라 하고, $\mathbb{C}$에 속한 참여자의 인덱스 집합을 $PLIST$로 가정하자. 즉 , $P_1$와 $P_3$가 참여할 경우 $\mathbb{C} = \{P_1, P_3\}$, $PLIST = \{1,3\}$ 이 된다.

$\mathbb{C}$에 속하는 각 참여자 $P_i$는 $PLIST$를 공유 하고 다음과 같은 Lagrange 베이시스 다항식(Lagrange basis polynomial)을 계산한다:

$$l_i(x) = \prod_{\substack{j \in PLIST \\ j \neq i}} \frac{x-j}{i-j} \pmod q$$

주목할 점은 각 참여자의 Lagrange 베이시스 다항식은 $PLIST$를 알고 있으면 누구나 계산할 수 있다는 점이다. 비밀정보를 복원하는 사람은 참여자들의 Lagrange 베이시스 다항식과 참여자들의 샤미르 비밀 공유를 선형 결합(linear combination)하여 공유 값 분배 시 사용한 다항식 $f(x)$를 복원할 수 있다. 또한 $f(0)$를 계산해서 비밀정보 $s$를 얻을 수 있다.

$$f(x) = \sum_{i \in PLIST} x_i l_i(x) \pmod q$$

$f(0)$를 풀어서 정리하면 다음과 같다:

$$f(0) = \sum_{i \in PLIST} x_i l_i(0) \pmod q$$
$$= \sum_{i \in PLIST} [x_i \prod_{\substack{j \in PLIST \\ j \neq i}} \frac{-j}{i-j}] \pmod q$$

$PLIST$가 공개된 상태에서 $\mathbb{C}$에 속하는 참여자 $P_i$는 자체적으로 $l_i(0)$를 계산하는 것이 가능하다. 따라서 $\mathbb{C}$에 속하는 모든 참여자가 자체적으로 $w_i = x_i l_i(0)$를 계산하면 $\{w_i\}_{i \in PLIST}$는 $s$의 additive 공유 값이 된다. 즉 $s = \sum_{i \in PLIST} w_i$ 가 된다. $TransStoA(i, PLIST)$를 아래와 같이 정의한다:

$$TransStoA(i, PLIST) = l_i(0) \pmod q$$
$$= \prod_{\substack{j \in PLIST \\ j \neq i}} \frac{-j}{i-j} \pmod q$$

### 2.4 Verifiable Secret Sharing

Feldman의 검증 가능한(verifiable) 비밀 공유[19]에서 딜러는 각 참여자 $P_i$에게 $f(i) \pmod q$를 보내는 것 외에 모든 참여자에게 아래의 부가적인 정보를 전달한다. 표기법은 위의 샤미르 비밀 공유 수식을 따른다.

$$V_j = a_j G \text{ for } all \; j \in \{0,1,...,t\}$$

$V_0$은 딜러가 분배하는 대상 비밀정보 $s$에 타원곡선 군의 생성자 $G$를 곱한 값이다. $P_i$는 아래의 식으로 자신의 전달 받은 공유 값 $f(i)$를 검증한다:

$$f(i)\, G \equiv \sum_{j=0}^{t} i^j V_j$$

만약 수식이 성립하지 않으면 비밀 공유 프로토콜을 종료한다.

### 2.5 비대화형 Schnorr 영지식 증명

증명자(prover)가 discrete log에 대한 지식을 검증자(verifier)에게 증명하기 위한 대표적인 기법은 Schnorr 프로토콜이 있으며[20], 여기에 해시함수를 적용한 Fiat-Shamir 기법[21]을 적용하여 비대화

형(non-interactive)으로 구성할 수 있다. 이를 타원곡선 군에 적용한 증명생성 함수 $proveZKDL$과 증명검증 함수 $verifyZKDL$의 알고리즘은 다음과 같다:

$proveZKDL(a, A = aG)$ :
$r \in_R Z_q$
$R_A = rG$
$c = Hash(R)$
$s_A = r + ca$
**return** $(R_A, s_A)$

$verifyZKDL(A, R_A, s_A)$ :
**if** $s_A G \equiv R_A + Hash(R_A)A$
**return** $true$
**else**
**return** $false$

증명자는 '타원곡선 군 $A$의 타원곡선 discrete log인 $a$를 알고 있음'에 대한 증명 $(R_A, s_A)$를 $proveZKDL(a, A)$를 통해서 생성하고 검증자는 $verifyZKDL(A, R_A, s_A)$를 통해서 증명을 검증한다.

## III. 관련 연구

DKG에 참여하는 참여자의 인덱스를 $PARTIES$라고 하자.

$(t,n)$ 임계값 ECDSA의 DKG 프로토콜[10]은 다음과 같으며 결과물로 각 참여자들은 $(t,n)$ 샤미르 비밀 공유 값과 공개키 $pk$를 가지게 된다:

**Phase 1**. 각 참여자 $P_i$는 난수 $u_i$를 선정하고, $U_i = u_i G$의 commitment를 다른 참여자들에게 전달한다. $u_i$는 $sk = \sum_{k \in PARTIES} u_k$인 additive 공유 값이 되고 $sk$는 암묵적으로 결정된다.

**Phase 2.** 각 참여자 $P_i$는 $U_i$의 decommitment를 다른 참여자들에게 전달하고, 각 참여자들은 다른 각 참여자 $P_j$의 $U_j$를 decommit 한다.

각 참여자 $P_i$는 자신의 비밀 정보인 $u_i$에 대해 랜덤한 다항식 $f_i(x)$를 만들고 Feldman의 검증 가능한 $(t,n)$ 비밀 공유를 수행한다. 즉 다른 참여자 $P_j$에게 $f_i(j)$를 비밀 채널로 송부하고 부가정보인 $V_{i,0}, V_{i,1}, ..., V_{i,t}$를 모든 참여자에게 공개 채널로 전달한다. 이 때 공개하는 $V_{i,0}$는 $U_i$가 된다.

각 참여자 $P_i$는 다른 참여자 $P_j$가 보낸 $V_{j,0}$와 decommit한 $U_j$가 동일한지 확인한다. 동일하지 않을 경우 프로토콜을 중지한다.

각 참여자 $P_i$는 $(t,n)$ 샤미르 비밀 공유 값 $x_i (= \sum_{k \in PARTIES} f_k(i))$를 가지게 되고, Feldman의 검증 가능한 비밀 공유의 부가정보를 이용해서 다른 각 참여자 $P_j$의 공유 값 $x_j$에 $G$를 곱한 $X_j$와 $pk (= \sum_{k \in PARTIES} V_{k,0})$를 알게 된다.

**Phase 3.** 각 참여자 $P_i$는 $proveZKDL(x_i, X_i)$를 실행하여 $\left(R_{X_i,i}, s_{X_i,i}\right)$를 계산한 후 다른 참여자들에게 전송한다. 이후 각 참여자 $P_i$는 다른 각 참여자 $P_j$에 대해 $verifyZKDL(X_j, R_{X_j,j}, s_{X_j,j})$를 실행한다. 실행결과가 false일 경우 프로토콜을 중지한다.

임계값 ECDSA 방법론마다 위의 프로토콜에 부가적으로 Paillier 반동형암호[22]를 생성하기도 한다[10,11]. 그러나 이는 디지털 서명 프로토콜을 위한 준비 작업을 병렬로 처리하기 위한 것으로 DKG 프로토콜의 정확성이나 보안성과는 상관이 없다. 대부분의 임계값 ECDSA 방법론들은 위 내용과 유사한 DKG 프로토콜을 적용한다[10,11,12,13].

임의의 $t+1$ 참여자들이 개인키 $sk$를 복원하지 않고 디지털 서명을 생성하는 서명 프로토콜이 임계값 ECDSA의 또 다른 주제이다. 본 논문은 서명 프로토콜을 가능하게 하는 샤미르 비밀 공유 값을 복구하는 것을 목표로 하므로 서명 프로토콜을 다루지는 않는다.

그러나 안전한 서명 프로토콜을 수행하기 위한 기존 연구의 DKG 보안 요구사항으로부터 본 논문의 보안 요구사항을 도출한다. 기존 연구의 $(t,n)$ DKG 프로토콜은 정확성과 기밀성 측면에서 다음의 공통적인 보안 요구사항을 만족한다[7]:

1) 정확성(Correctness) : 모든 $(t,n)$ 공유 값 중 임의의 $t+1$ 공유 값으로 복원되는 개인키는 동일한 $sk$이다(C1). 모든 참여자는 동일한 $pk (= skG)$를 가진다(C2). DKG를 통해서 생성되는 $sk$는 $Z_q$에서 균등분포를 가진다(uniformly distributed)(C3).

2) 기밀성(Secrecy) : $t$명 이하의 참여자로는 $sk$를 추론할 수 없다(S1).

프로토콜 수행 후 참여자가 보유하게 되는 정보를 뷰(view)라고 한다. 상기 DKG 프로토콜에서 참여자가 프로토콜 수행 후 보유하는 뷰는 다음과 같다:

$$View_i^{DKG} = \{u_i, f_i(x), x_i, pk\} \cup \{U_j, f_j(i), V_{j,0}, V_{j,1}, ..., V_{j,t}, R_{X_j}, s_{X_j}\} \text{ for } j \in PARTIES$$

[7]은 Phase 1에서 $sk$가 암묵적으로 결정된 상태에서 프로토콜을 진행할 경우 각 참여자 $P_i$는 $View_i^{DKG}$로부터 $sk$를 추론할 수 없음을 증명하고 있다.

## IV. 제안 프로토콜

### 4.1 보안 요구사항

본 논문에서 제안하는 복구 프로토콜의 보안 요구사항은 DKG 프로토콜의 보안 요구사항으로부터 도출된다. DKG 프로토콜과는 달리 복구 프로토콜에서는 개인키를 새로 생성하는 것이 아니고 기존에 존재하는 고정된 개인키를 대상으로 하고 있다. 따라서 복구 프로토콜에서는 요구사항 (C1)과 (C2)는 유지되고, 새로운 개인키 생성을 감안한 요구사항인 (C3)는 제외된다. 또한 본 논문에서는 (1,3) 샤미르 비밀 공유를 대상으로 한다. 이에 적합한 새로운 보안 요구사항은 다음과 같다:

1) 정확성 : 모든 공유 값 중 임의의 2개 공유 값으로 기존 개인키인 $sk$를 복원할 수 있다(C1). 모든 참여자는 기존 공개키와 동일한 $pk(=skG)$를 가진다(C2).

2) 기밀성 : 참여자 단독으로는 $sk$를 추론할 수 없다(S1).

### 4.2 복구 프로토콜

공유 값을 분실한 참여자는 $P_1$로, 공유 값을 보관하고 있는 참여자는 $P_2$와 $P_3$으로 가정한다.

주목할 점은 $P_2$와 $P_3$의 경우 DKG 프로토콜을 통해서 각각 $x_2$와 $x_3$를 가지고 있고, $X_1$, $X_2$, $X_3$, $pk$를 공유하고 있다. DKG 프로토콜을 확장한 복구 프로토콜은 다음과 같다. Fig. 1은 복구 프로토콜의 Phase 1과 Phase2를 도식화하여 보여준다.

**Phase 1.** $P_2$와 $P_3$는 다음 연산을 로컬에서 수행한다 ($i \in \{2,3\}$):

$$u_i \xleftarrow{\$} Z_q$$
$$w_i := x_i \, TransStoA(i,(2,3)) \pmod q$$
$$k_i := w_i - u_i \pmod q$$
$$U_i := u_i G$$
$$R_{W_i}, s_{W_i} := proveZKDL(w_i, W_i)$$
$$W_j := X_j \, TransStoA(j,(2,3)) \quad where\ j = 3 \text{ if } i = 2 \text{ or } j = 2 \text{ if } i = 3$$
$$pk_i := pk$$

$P_2$와 $P_3$는 다음 정보들을 $P_1$에게 비밀 채널로 전달한다 ($i \in \{2,3\}$):

$$k_i,\ U_i,\ R_{W_i},\ s_{W_i},\ W_{5-i}, pk_i$$

**Phase 2.** $P_1$은 아래 조건들을 체크하고 하나 이상의 조건이 충족되지 않을 경우 중지한다:

$$pk_2 == pk_3$$
$$W_2 \equiv U_2 + k_2 G$$
$$W_3 \equiv U_3 + k_3 G$$
$$W_2 + W_3 \equiv pk_2$$
$$verifyZKDL(W_2, R_{W_2}, s_{W_2})$$
$$verifyZKDL(W_3, R_{W_3}, s_{W_3})$$

$u_1$을 $k_2 + k_3 \pmod q$로, $pk_2$를 $pk$로 지정한다.

**Phase 3.** 각 참여자 $P_i$는 자신의 비밀 정보인 $u_i$에 대해 랜덤한 1차 다항식 $f_i(x)$를 만들고

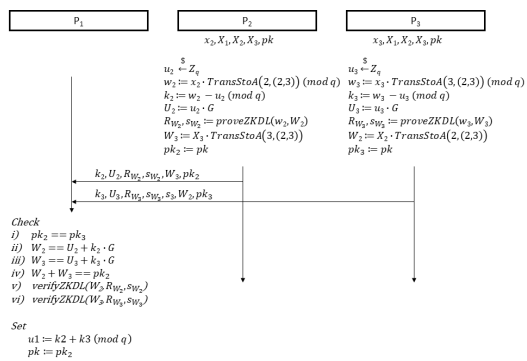


Fig. 1. Phase 1 and Phase 2 Processes.

Feldman의 검증 가능한 $(t,n)$ 비밀 공유를 수행한다. 즉 각 참여자 $P_j$에게 $f_i(j)$를 비밀 채널로 송부하고 부가정보인 $V_{i,0}$과 $V_{i,1}$를 모든 참여자에게 공개 채널로 송부한다. 이 때 공개하는 $V_{i,0}$는 $U_i$가 된다.

각 참여자 $P_i$는 다른 참여자 $P_j$가 보낸 $V_{j,0}$의 합과 자신의 $V_{i,0}(=U_i)$를 합하여 $pk$가 나오는지 확인한다. 동일하지 않을 경우 프로토콜을 중지한다.

각 참여자 $P_i$는 $(t,n)$ 샤미르 비밀 공유 값 $x_i(=\sum_{k=1}^{3} f_k(i))$를 가지게 되고, Feldman의 검증 가능한 비밀 공유의 부가정보를 이용해서 다른 각 참여자 $P_j$의 공유 값 $x_j$에 $G$를 곱한 $X_j$를 알게 된다. 이후 $pk$와 $\sum_{k=1}^{3} V_{k,0}$이 같은지를 검증한다. 검증에 실패하면 중지한다.

**Phase 4.** 각 참여자 $P_i$는 $proveZKDL(x_i, X_i)$를 실행하여 $(R_{X_i}, s_{X_i})$를 계산한 후 다른 참여자들에게 전송한다. 이후 각 참여자 $P_i$는 다른 참여자 $P_j$에 대해 $verifyZKDL(X_j, R_{X_j}, S_{X_j})$를 실행한다. 실행 결과가 false일 경우 프로토콜을 중지한다.

제안 프로토콜에서는 commitment scheme을 생략한다. DKG 프로토콜에서 commitment scheme을 사용한 것은 특정 참여자가 다른 참여자의 타원곡선 군 원소 $U_j$를 받은 후 자신의 $u_i$와 $U_i$를 조정해서 공개키 $pk$의 분포에 영향을 줄 수 있기 때문이다[7]. 그러나 제안 프로토콜에서는 $pk$가 이미 결정되어 있고 Phase 1에서 결정된 $u_i$와 $U_i$에 대해 Phase 3에서 이미 알고 있는 $pk$와 $\sum_{k=1}^{3} U_k$가 같은지를 모든 참여자가 검증할 수 있기 때문에 commitment scheme을 생략할 수 있다.

## V. 보안 요구사항 검증

본 장에서는 앞에서 제안한 프로토콜이 보안 요구사항을 충족함을 검증한다.

1) C1의 증명: 세 참여자가 Phase 3에서 랜덤하게 생성한 1차 다항식을 합해서 다음의 1차 다항식을 정의하자:

$$f(x) = f_1(x) + f_2(x) + f_3(x)$$

이때 $f_1(0) + f_2(0) + f_3(0) = u_1 + u_2 + u_3 = sk$가 된다. $P_i$가 보유한 샤미르 시크릿 공유 값은 $x_i(=\sum_{k=1}^{N} f_k(i)) = f(i)$이므로 이중 두개의 공유 값으로 Lagrange 보간법을 통해서 동일한 $f(x)$와 $f(0) = sk$를 복원할 수 있다. 만약 $P_i$가 $P_j$에게 잘못된 $f_i(j)$를 보내서 $x_j$가 잘못된 공유 값을 가질 경우에는 Phase 3의 Feldman 공유 검증 단계에서 발견되고 프로토콜이 종료된다. 따라서 프로토콜이 정상적으로 완료되었다면 모든 $x_i$는 $f(x)$에 대해 적합한 샤미르 비밀 공유 값이 된다.

2) C2의 증명: 모든 참여자들은 모든 $i \in PARTIES$에 대해 $V_{i,0}$과 $V_{i,1}$를 알고 있으므로 동일한 $f_i(x)G$를 복원하는 것이 가능하고 $f(x)G$ 역시 알 수 있다. 따라서 $f(0)G$를 알 수 있다. 기존 $pk$와 다를 경우 Phase 3에서 프로토콜이 중지된다.

3) S1의 증명: 본 증명은 다음과 같이 진행된다. 복구 프로토콜에서 각 참여자 $P_i$가 확보하는 뷰를 $View_i^{Recovery}$라고 하자. 그리고 복구 과정에서 추가되는 정보 $View_i^{Add}$를 다음과 같이 정의한다:

$$View_i^{Add} = View_i^{Recovery} - View_i^{DKG}$$

먼저 $View_i^{Add}$만으로는 $sk$를 추론할 수 없음을 보이고, $View_i^{Add}$와 $View_i^{DKG}$를 합쳐도 $sk$에 대한 정보를 얻을 수 없다는 것을 설명한다. 결과적으로 $View_i^{DKG}$로 $sk$를 추론할 수 없다는 [7]의 증명에 따라 $View_i^{Recovery}$로는 $sk$를 추론할 수 없게 된다.

복구 프로토콜을 통해서 각 참여자가 얻게 되는 정보를 정리해 보자.

**Phase 1.**

$P_1$ : $k_2, k_3, U_2, U_3, W_2, R_{W_2}, s_{W_2}, W_3, R_{W_3}, s_{W_3}, pk$

$P_i$ : $u_i$ $(i \in \{2,3\})$

**Phase 2.**

$P_1$ : $u_1$

**Phase 3.**

$P_i$ : $f_i(x), x_i, U_j, f_j(i),$

$$V_{j,0},\ V_{j,1},\ X_j\ (j\in\{1,2,3\})$$

**Phase 4**.

$P_i$: $R_{X_j}, s_{X_j}(j\in\{1,2,3\})$

모든 $P_i$는 다음과 같은 $View_i^{DKG}$를 공통으로 가진다:

$$View_i^{DKG}=\{u_i,f_i(x),x_i,pk\}\cup \{U_j,f_j(i),V_{j,0},V_{j,1},R_{X_j},s_{X_j}\} \text{ for } j\in PARTIES$$

복구 프로토콜을 완료한 후 $P_1$, $P_2$, $P_3$가 가지는 뷰는 다음과 같다:

$$View_1^{Recovery}=View_1^{Add}\cup View_1^{DKG}$$
$$where\ View_1^{Add}=\{k_2,k_3,W_2,R_{W_2},s_{W_2}W_3,R_{W_3},s_{W_3}\}$$
$$View_2^{Recovery}=View_2^{DKG}$$
$$View_3^{Recovery}=View_3^{DKG}$$

$sk=k_2+k_3+u_2+u_3$이고 $u_2$와 $u_3$는 $P_2$와 $P_3$가 랜덤하게 선정한 숫자이므로 $P_1$은 $k_2$와 $k_3$으로부터 $sk$를 추론할 수 없다. 또한 $pk=W_2+W_3$이다. ECDLP에 의해 $W_2(=w_2G)$와 $W_3(=w_3G)$로부터는 $sk(=w_1+w_2)$를 추론할 수 없다. 만약 가능하다면 다음 연산이 가능하다. $kG$가 있을 때 $c\in Z_q$를 선택한 후 $kG$와 $cG$로부터 $(k+c)$를 추출한 후 $(k+c)$-$c$ (mod $q$)로 $k$를 알 수 있다. 따라서 ECDLP 가정에 위배된다. $R_{W_2}, s_{W_2}$는 $W_2$에 대한 지식을, $R_{W_3}, s_{W_3}$는 $W_3$에 대한 지식을 서로 다른 주체($P_2$와 $P_3$)가 독립적으로 영지식 증명한 것이다. 따라서 영지식 증명 정의상 $P_1$은 $w_2$와 $w_3$를 추론할 수 없고 따라서 $sk$를 알 수 없다.

$P_1$은 $View_1^{Add}$를 Phase 1에서 알게 되고 $pk$와 $U_j$를 제외한 $View_1^{DKG}$는 Phase 2 이후 알게 된다. 또한 $pk$와 $U_j$를 제외한 $View_1^{DKG}$는 $P_2$와 $P_3$가 Feldman 비밀 공유를 위해 선택하는 랜덤 1차 다항식, $f_2(x)$와 $f_3(x)$에 의해 결정된다. 그리고 $f_2(x)$와 $f_3(x)$는 각 다항식의 1차 계수에 의해 $View_1^{Add}$에 독립적이다. 따라서 $P_1$은 $View_1^{Add}$를 알고 있어도 $View_1^{DKG}$로 $sk$를 추론할 수 없다.

## VI. 복구 프로토콜 계산량 측정

블록체인에서 가장 많이 사용되는 ECDSA secp256k1 타원곡선을 이용한 비대칭키 시스템 상의 샤미르 공유 값 복원 과정을 Golang으로 개발하여 실행 속도를 측정하였다. Fig. 2에서 보듯이 제안 프로토콜의 실제 어플리케이션 적용 시 속도를 추정하기 위해 참여자간의 네트워킹(all-parties-connection), 복구 프로토콜(share recovery), 임계값 서명을 위한 준비 과정(sign preparation), 공유 값 저장 및 확인(all-parties save and confirm) 과정을 포함하여 구현하고 100번 반복 실행한 후 단계별 평균 속도를 분석했다. AMD Ryzen Pro 7과 16GB 메모리의 PC에서 $P_2$과 $P_3$에 해당하는 서버는 도커(docker)로, $P_1$은 클라이언트 프로그램으로 작동되었다. 이들 간의 통신은 Web Socket으로 이루어졌다. 본 프로토콜에서 제안된 복구 프로토콜은 평균적으로 약 9ms 이하의 속도를 보이고 있음을 확인할 수 있었다. 다만 임계값 서명을 위한 준비 과정에서 Paillier 반동형암호를 이용하므로 다소 긴 시간이 소요되고 있다. 이는 일회성 연산이므로 이후의 임계값 서명 프로토콜 수행 시 다시 수행될 필요가 없다.

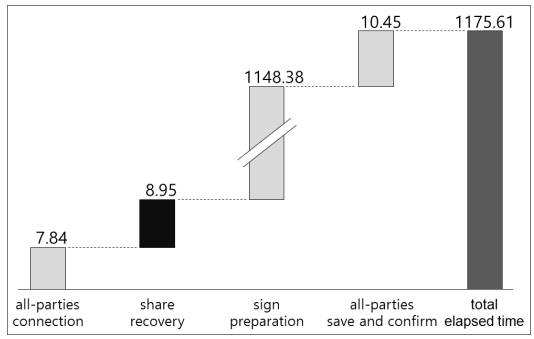


Fig. 2. Analysis of Share Recovery Time (milli-seconds, 100 iteration).

## VII. 적용 가능한 블록체인 기반 가상자산 관리 영역

최근 국내외 가상자산 거래소 혹은 관련 기업에서 발생하는 대량의 가상 자산 유출 사고들은 대부분 블록체인 개인키의 유출로 발생하고 있다. 본 논문에서 제안된 (1,3) 분산키의 복구 프로토콜은 이러한 어

려움을 해소하기 위한 방법으로써 블록체인 기반 가상 자산 관리를 위한 안전한 개인키 관리와 서명 권한 위임 및 불능화(revocation)에 적용 가능하다.

먼저 기업의 가상 자산 관리를 위한 안전한 개인키 관리에 적용될 수 있다. 자산을 관리하기 위해서는 담당 직원의 디바이스 혹은 관리 서버가 개인키를 보유하고 디지털 서명을 생성하게 된다. 이때 담당 직원의 디바이스 혹은 관리 서버를 해킹하면 바로 개인키를 취득할 수 있기 때문에 단일 실패 지점이 존재하게 된다.

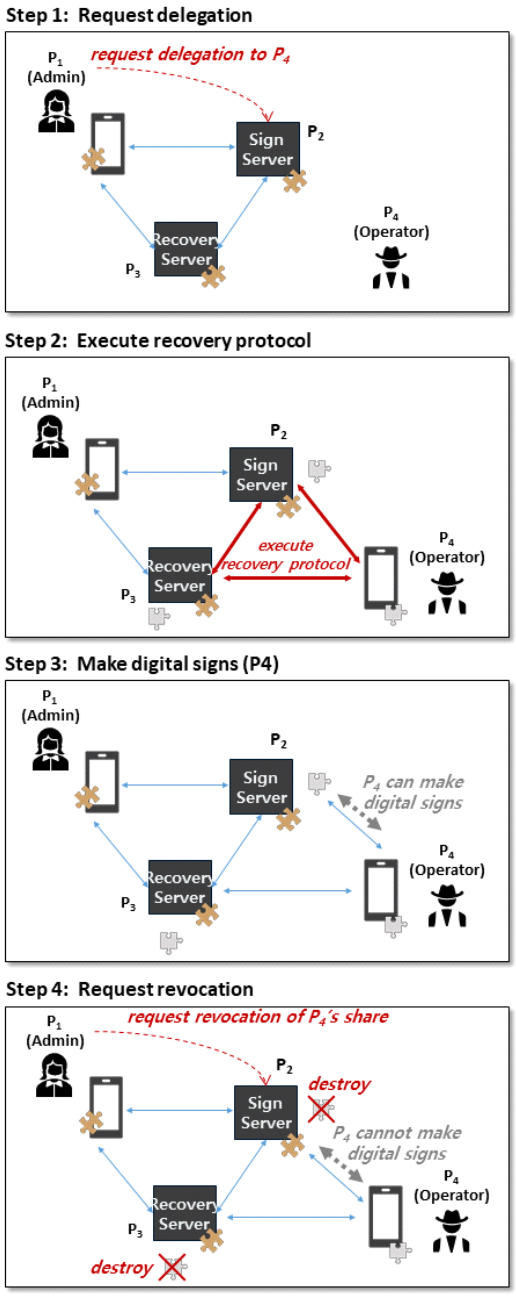


Fig. 3. Delegation and Revocation Processes.

$P_1$을 담당 직원, $P_2$를 서명 서버, $P_3$를 복구 서버로 지정하고 이들에게 $(1,3)$ 분산키를 분배하고 일상적인 디지털 서명은 $P_1$과 $P_2$가 임계값 서명을 하고 $P_3$는 복구 시에만 온라인 접속이 가능하게 한다. 이후 담당 직원 혹은 한 서버의 공유 값이 유실되거나 해킹이 의심될 경우 본 논문의 복구 프로토콜을 통해서 삼자가 새로운 공유 값을 갖도록 한다. 이때 기존 공유 값들은 삭제한다. 이후 해킹된 공유 값으로는 임계값 서명을 할 수 없게 된다.

두 번째로는 블록체인 기반 가상 자산에 대한 디지털 서명 권한의 위임과 불능화가 가능해진다. 특히 공개 블록체인에서는 은행이나 공공 기관이 개입하여 자산의 이동을 통제하는 것이 불가능하고 개인키를 보유한 사람은 언제든 어느 장소에서든 자산을 이동시키는 것이 가능하다. 그러므로 개인키를 위임자에게 넘겨 준 이후에는 위임자의 서명 권한을 완전하게 불능화시키는 것은 검증이 불가능하다.

Fig. 3은 위 사례를 확장하여 서명 권한을 위임하고 불능화 하는 과정을 보여준다. $P_1$이 다른 직원 $P_4$에게 한시적으로 서명 권한을 위임한다고 가정해 보자(Step 1). 이때 $P_2$와 $P_3$은 $P_1$의 요청에 의해 $P_4-P_2-P_3$간의 복구 프로토콜을 진행한다(Step 2). 이후 $P_4$는 $P_2$와 임계값 서명을 수행할 수 있다(Step 3). 위임 기간이 종료되면 $P_2$와 $P_3$은 $P_4$와의 복구 프로토콜에서 생성된 공유 값들을 삭제함으로써 $P_4$가 가진 공유 값을 이용한 서명 기능을 불능화시킬 수 있다(Step 4).

## VIII. 결 론

비신뢰 기반 DKG 프로토콜과 임계값 서명 프로토콜은 개인키 분실 및 유출의 단일 실패 지점을 제거함으로써 개인키 관리와 활용의 안전성을 높인다. 최근 블록체인 가상 자산의 활용 범위와 가치가 증가함에 따라 관련 이론과 기술이 급속도로 발전하고 있다.

본 논문에서는 이 두 프로토콜과 연계하여 유용하게 적용 가능한 분산키의 복구 프로토콜을 $(1,3)$ 분

산키에 대해 제안하였다. 비신뢰 기반 DKG 프로토콜로 생성된 (1,3) 분산키 중 한 개가 유실될 경우에도 남은 두 개의 공유 값으로 임계값 서명이 가능하다. 그러나 이중 하나가 다시 유실될 경우 개인키를 복원하거나 임계값 서명을 하는 것이 불가능해진다. 따라서 빠른 시간에 복구 프로토콜을 적용하여 같은 개인키에 대해 새로운 (1,3) 공유 값을 재분배하는 것이 필요하다.

제안된 복구 프로토콜의 첫 단계에서는 공유 값을 보유한 두 참여자가 공유 값를 분실한 참여자의 난수 생성을 지원한다. 이후 세 참여자는 축약된 DKG 프로토콜을 수행한다. 복구 프로토콜은 정확성과 기밀성 측면에서 DKG 프로토콜과 동일한 수준의 보안 요구사항을 만족한다.

본 연구결과는 기업의 블록체인 기반 가상 자산 관리 분야에서 안전한 개인키 관리와 서명 권한 위임 및 불능화를 가능하게 함으로써 보다 안전한 자산 관리에 기여할 수 있을 것으로 기대된다.

(1,3) 분산키가 유용한 구조이기는 하지만 보다 넓은 활용성을 가지기 위해서 일반적인 $(t,n)$ 분산키 복구에 활용 가능한 확장된 복구 프로토콜을 추가적으로 연구할 계획이다.

## References


[1] Dekey Wallet | dApp wallet, "Dekey", https://dekey.app/, 2021.08.16.

[2] MetaMask - A crypto wallet & gateway to blockchain apps, "Metamask", https://metamask.io/, 2021.08.16.

[3] Ledger: Hardware Wallet, "Ledger", https://www.ledger.com/, 2021.08.16.

[4] D'CENT Wallet, "DCent", https://dcentwallet.com/, 2021.08.16.

[5] Shamir, A., "How to share a secret." Communications of the ACM vol.22, no.11, pp.612-613, Nov, 1979.

[6] Pederson, T.P., "Non-interactive and information-theoretic secure verifiable secret sharing," Annual international cryptology conference, pp. 129-140, Aug, 1991.

[7] Gennaro, R., Jarecki, S., Krawczyk, H., and Rabin, T., "Secure distributed key generation for discrete-log based cryptosystems," International Conference on the Theory and Applications of Cryptographic Techniques, pp. 295-310, May, 1999.

[8] Certicom, Sec 1: Elliptic curve cryptography, Certicom Research v2, 137, 2009.

[9] Cramer, R., Damgård, I., and Maurer, U., "General secure multi-party computation from any linear secret-sharing scheme," International Conference on the Theory and Applications of Cryptographic Techniques, pp. 316-334, May, 2000.

[10] Gennaro, R., and Goldfeder, S., "Fast multiparty threshold ECDSA with fast trustless setup," Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security, pp. 1179-1194, Oct, 2018.

[11] IACR epring archieve: Gennaro, R., and Goldfeder, S, "One round threshold ECDSA with identifiable Abort," IACR ePring, 2020-540, May, 2020.

[12] Canetti, R., Gennaro, R., Goldfeder, S., Makriyannis, N. and Peled, U., "UC non-interactive, proactive, threshold ECDSA with identifiable aborts," Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, pp. 1769-1787, Oct, 2020.

[13] Damgård, I., Jakobsen, T. P., Nielsen, J. B., Pagter, J. I., and Østergård, M., "fast threshold ECDSA with honest majority," International Conference on Security and Cryptography for Networks, pp. 382-400, Sep, 2020.

[14] BitGo: Institutional Digital Asset

Platform, “Bitgo”, https://www.bitgo.com/services/custody/wallet-platform/, 2021.08.16.
[15] Fortmatic, “Security & infrastructure at fortmatic”, https://medium.com/fortmatic/security-infrastructure-at-fortmatic-4a95c3688997. 2021.08.16.
[16] AWS documentation, “AWS encryption sdk”, https://docs.aws.amazon.com/encryption-sdk/latest/developer-guide/introduction.html, 2021.08.16.
[17] Microsoft Ingite, “Azure key vault rest api reference”, https://docs. microsoft.com/en-us/rest/api/keyvault/, 2021.08.16.
[18] Waring, E., “Vii. problems concerning interpolations,” Philosophical transactions of the royal society of London 69, pp. 59-67, Jan, 1779.
[19] Feldman, P., “A practical scheme for non-interactive verifiable secret sharing,” 28th Annual Symposium on Foundations of Computer Science, pp. 427-438, Oct, 1987.
[20] Schnorr, C. P., “Efficient signature generation by smart cards,” Journal of cryptology, vol. 4, no. 3, pp. 161-174, Jan, 1991.
[21] Fiat, A., and Shamir, A., “How to prove yourself: Practical solutions to identification and signature problems,” Conference on the theory and application of cryptographic techniques, pp. 186-194, Aug, 1986.
[22] Paillier, P., “Public-key cryptosystems based on composite degree residuosity classes,” International conference on the theory and applications of cryptographic techniques, pp. 223-238, May, 1999.

## 〈저 자 소 개〉

배 경 일 (Kyoungil Bae) 정회원
1994년 2월: 한국과학기술원 수학과 학사
1996년 2월: 한국과학기술원 산업경영학과 석사
2002년 2월: 한국과학기술원 경영공학과 박사
2002년 4월~2005년 7월, 2011년 11월~2014년 2월: 한국IBM
2014년 2월~2016년 6월: 포스코기술투자
2018년 10월~현재: ㈜아톰릭스랩
〈관심분야〉 블록체인, 암호학, 금융공학

박 준 후 (Junhoo Park) 정회원
2014년 2월: 충남대학교 정보통신공학과 학사
2016년 2월: 충남대학교 컴퓨터공학과 석사
2016년 3월~현재: 충남대학교 컴퓨터공학과 박사과정
2020년 1월~현재: ㈜아이오트러스트
〈관심분야〉 블록체인, 보안 프로토콜, 암호 응용

류 재 철 (Jaecheol Ryou) 종신회원
1985년 2월: 한양학교 산업공학과 졸업
1988년 5월: Iowa State University 전산학 석사
1990년 12월: Northwestern University 전산학 박사
1991년 2월~현재: 충남학교 컴퓨터공학과 교수
〈관심분야〉 모바일 보안, 금융보안, 블록체인