사물인터넷융복합논문지 Vol. 12, No. 5, pp. 141-148, 2026 https://doi.org/10.20465/KIOTS.2026.12.5.141

AI 에이전트 기반 다자간 회의를 위한 Human-Agent 동적 그룹 공동서명 기법

윤성현*
백석대학교 컴퓨터공학부 교수

A Human-Agent Dynamic Group Joint Signature Scheme for Multi-Party Meetings with AI Agents

Sung-Hyun Yun*
Professor, Division of Computer Engineering, Baekseok University

요 약 생성형 AI의 발전으로 사람과 AI 에이전트가 함께 참여하는 다자간 회의 환경이 요구되고 있다. 기존 회의 인증 방식만으로는 특정 의사결정에 어떤 사람과 AI 에이전트가 참여했는지와 해당 의사결정 당시의 참여자 구성이 이후에 변조되지 않았는지를 함께 검증하기 어렵다. 본 논문에서는 HP(Human Participant)와 AA(AI Agent)를 서로 다른 방식으로 인증한 후 동일한 Schnorr 기반 공동서명에 참여하는 Human-Agent 동적 그룹 공동서명 기법을 제안한다. HP는 FE(Fuzzy Extractor)와 난수를 이용해 키를 생성하고, AA는 전용 개인키와 식별자, 소유자, 역할, 모델 정보를 포함하는 인증서를 사용한다. 회의 진행 중 가입과 탈퇴가 발생할 때마다 공동 공개키, 회의 참여자 목록의 Merkle root, 그룹 버전 및 이전 상태를 결합한 그룹 상태값을 생성하여 블록체인에 저장한다. 또한 공동서명을 메시지, 회의 식별자, 의사결정 식별자 및 그룹 상태에 결합함으로써 다른 회의나 변경된 그룹 상태에서 기존 서명이 재사용되는 것을 방지한다. 제안한 기법은 사람과 AI 에이전트가 함께 참여하는 회의에서 공동 의사결정의 무결성과 사후 검증 가능성을 제공한다.

주제어 : AI 에이전트, Human-Agent 협업, 공동서명, Schnorr 다중서명, 동적 그룹, 블록체인 감사

Abstract With the rapid advancement of generative AI, multi-party meeting environments in which human participants and AI agents jointly participate in decision-making are increasingly required. However, conventional meeting authentication mechanisms have limitations in simultaneously verifying which humans and AI agents participated in a specific decision and whether the participant composition recorded for the decision has been subsequently altered. This paper proposes a Human-Agent Dynamic Group Joint Signature scheme in which Human Participants (HPs) and AI Agents (AAs) are authenticated using different mechanisms and then jointly participate in the same Schnorr-based joint signature process. An HP generates a cryptographic key using a Fuzzy Extractor (FE) and a random value, whereas an AA is authenticated using a dedicated private key and a certificate containing its identifier, owner, role, and model information. Whenever a participant joins or leaves during a meeting, a group state value is generated by combining the aggregated public key, the Merkle root of the participant list, the group version, and the previous group state, and the resulting state is recorded on a blockchain. In addition, the joint signature is bound to the message, meeting identifier, decision identifier, and group state, thereby preventing an existing signature from being reused in another meeting or under a modified group configuration. The proposed scheme provides integrity and post-event verifiability of collective decisions in meeting environments where humans and AI agents jointly participate.

Key Words : AI Agent, Human-Agent Collaboration, Joint Signature, Schnorr Multi-signature, Dynamic Group, Blockchain Audit

본 논문은 2018년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임(2018R1D1A3B07050180).
*교신저자 : 윤성현(shcrpt@gmail.com)
접수일 2026년 09월 29일 수정일 2026년 10월 01일 심사완료일 2026년 10월 18일

www.kci.go.kr

1. 서론

대규모 언어모델을 기반으로 한 AI 에이전트는 단순한 대화 생성 기능을 넘어 외부 도구 호출, 정보 검색, 데이터 분석 및 작업 분담을 수행한다. AutoGen과 AutoAgents는 여러 AI 에이전트가 서로 다른 역할을 수행하고 사람과 상호작용하면서 복합적인 작업을 협력적으로 처리할 수 있음을 보여준다[1, 2]. 또한 생성형 에이전트 연구는 AI 에이전트가 독립적인 역할과 상태를 유지하면서 사람 또는 다른 에이전트와 협력할 수 있는 가능성을 제시한다[3, 4].

향후 다자간 회의에서는 AA(AI Agent)는 단순히 회의 내용을 요약하는 AI 도구뿐 아니라 특정 자료를 조사하고, 대안을 분석하며, 의사결정을 지원하거나 일부 승인 절차에 참여하는 실제 회의 구성원으로의 역할이 요구된다.

HP(Human Participant)와 AA가 함께 참여하는 회의에서 HP의 입장과 퇴장, AA의 추가와 제거, 역할 변경 등이 동적으로 발생할 수 있으며, 이에 따라 특정 시점의 참여자 구성과 권한도 지속적으로 변경된다. 기존의 바이오메트릭 기반 동적 그룹서명은 그룹 구성원의 가입과 탈퇴에 따라 그룹 공개값을 갱신하는 구조이다[5]. 하지만 사람 중심 환경을 대상으로 하기 때문에 AI 에이전트가 독립적인 인증 및 서명 주체로 참여하는 환경을 직접 지원하지 않는다.

더불어 최종 의사결정에 MM(Meeting Manager)이나 단일 서버의 전자서명만 남기는 경우 해당 의사결정 당시 실제로 어떤 HP와 AA가 참여하였는지와 각 참여자가 승인된 권한을 가지고 있었는지를 사후에 확인하기 어렵다. 모든 MP가 개별적으로 전자서명을 생성하면 서명 참여 사실을 개별적으로 검증할 수 있지만 MP 수에 비례하여 서명 데이터와 검증 비용이 증가한다. 본 논문에서 공동서명은 의사결정 결과에 대한 만장일치 찬성 투표를 의미하지 않는다. 공동서명은 회의 규칙에 따라 확정된 의사결정 기록 M의 내용, 참여자 구성 및 각 MP의 참여·의견이 정확히 기록되었음을 서명자 집합이 확인하는 증명이다. 따라서 반대 의견을 가진 MP도 자신의 반대가 M에 정확히 기록되어 있음을 확인한 후 공동서명에 참여할 수 있다.

따라서 현재 승인된 참여자 집합을 하나의 공동 공개키와 그룹 상태로 표현하고, 해당 서명자들이 확인한 의사결정 기록을 하나의 공동서명으로 검증하면서 구성 변경 이력도 추적할 수 있어야 한다.

본 연구에서 HP와 AA는 서로 다른 신원 특성을 가지므로 각 주체에 적합한 인증 절차를 적용하고, 인증을 통과한 이후에는 동일한 공개키 기반 서명 인터페이스를 사용하는 MP로 통합한다.

$$HP,\ AA \xrightarrow{Authentication} MP \xrightarrow{Signature} Joint\ Signature$$

HP의 바이오메트릭 정보는 FE를 이용하여 안정적으로 복구 가능한 비밀값을 생성하고[6], 등록 시 단말에서 생성한 난수와 결합하여 개인키를 생성한다.

AA는 안전한 난수를 이용하여 독립적으로 개인키를 생성하고, CA(Certification Authority)는 에이전트의 식별자, 공개키, 소유자, 역할 및 실행환경 정보를 결합한 인증서를 발급한다.

HP와 AA에 서로 다른 등록 및 인증 절차를 적용하되 인증 이후에는 동일한 MP로 추상화하여 하나의 공개키 기반 공동서명 프로토콜에 참여하도록 한다.

HP 또는 AA의 가입과 탈퇴가 발생할 때마다 현재 MP의 공개키 집합을 반영하여 공동 공개키 GPK_v를 갱신하고, HP와 AA의 구성을 각각 HR_v와 AR_v에 반영한다.

메시지와 MID, DecID, GID, 그룹 버전 v, 그룹 상태 GS_v 및 서명 세션 식별자 SID를 공동서명에 결합하여 이전 회의, 이전 그룹 상태 또는 다른 동시 서명 세션에서 생성된 공동서명의 재사용을 방지한다.

공동 공개키와 그룹 상태를 BC(Blockchain)에 버전별로 기록하여 특정 의사결정 시점에 어떤 HP와 AA가 승인된 MP로 참여하였는지를 사후 검증할 수 있도록 한다.

2장에서는 관련 연구를 살펴보고, 3장에서는 제안한 방법의 등록, 가입, 그룹 상태, 공동서명 생성 및 검증 절차를 설명한다. 4장에서는 공격자 모델, 안전성 및 정량적 효율성을 분석하고, 5장에서 결론 및 향후 연구 방향을 제시한다.

2. 관련 연구

2.1 그룹서명과 공동서명

Chaum과 van Heyst가 제안한 고전적 그룹서명은 그룹 구성원 한 명이 자신의 실제 신원을 공개하지 않고 그룹을 대표하여 서명할 수 있도록 한다[8]. 그룹서명의 익명성, 추적성 및 위조 불가능성 등에 대한 정의가 이후 연구에서 정립되었으며[9], 효율적인 그룹서명 기법들이

www.kci.go.kr

제안되었다[10, 11].

반면 공동서명은 서로 다른 개인키를 가진 복수의 참여자가 동일한 메시지 또는 동일한 기록에 대한 확인을 하나의 서명으로 결합하여 증명하는 것을 목적으로 한다. 일반적인 공동서명에서는 참여자 집합이 확인 가능하므로 익명성보다는 공동 참여 여부 검증과 서명 크기의 축소가 주요 목적이다.

본 연구의 공동서명 구조는 Schnorr 서명의 선형 결합 특성을 이용한다[12]. 여러 Schnorr 공개키를 결합하여 공동 공개키를 구성하고 부분서명을 결합하면 하나의 최종 서명을 생성할 수 있다.

하지만 공개키를 단순히 결합하면 rogue-key 공격에 취약하고 MuSig 연구에서는 공개키 집합을 반영한 키 계수와 nonce 처리 절차 등을 사용하여 이러한 공격에 대응한다[13, 14].

본 논문에서는 공개키에 대응하는 개인키의 보유 여부를 검증하기 위해 Schnorr 기반 PoP(Proof of Possession)를 사용하고, 공동서명 과정에서는 난수 값 공개 전에 커밋먼트를 제출하도록 구성한다. 각 MP는 가입 과정에서 PoP를 제출하여 등록한 공개키에 대응하는 개인키를 실제로 보유하고 있음을 증명한다. 또한 공동서명 과정에서는 난수 값을 바로 공개하지 않고 커밋먼트 값을 먼저 제출한 후 난수 값을 공개하도록 구성한다.

본 논문의 동적 그룹 공동서명은 승인된 HP와 AA의 공동 의사결정과 동적 구성 변화를 검증하는 것을 목적으로 한다.

2.2 바이오메트릭 인증과 디지털 신원

바이오메트릭 정보는 매번 측정할 때 마다 동일한 값이 생성되지 않으며, 일반적인 암호학적 난수와 비교하여 엔트로피가 제한될 수 있다. FE는 서로 가까운 바이오메트릭 입력으로부터 동일한 비밀값을 안정적으로 복구할 수 있도록 한다[6]. 등록 단계에서 비밀값과 공개 가능한 helper data를 생성하고, 재인증 과정에서는 새로운 바이오메트릭 입력과 helper data를 사용하여 동일한 비밀값을 복구한다.

NIST 디지털 신원 지침은 바이오메트릭 인증에서 인증장치, 재시도 제한 및 인증정보 보호 등의 보안 요구사항을 제시한다[7]. 바이오메트릭 원본은 한 번 도용되면 재사용할 수 없기 때문에 데이터베이스에 직접 저장하지 않는다.

W3C의 DID (Decentralized Identifier)는 사람, 조직, 장치 등 다양한 주체를 검증 가능한 디지털 식별자로 표현할 수 있는 구조를 제공한다[15]. DID는 AA의 디지털 신원을 구성하는 데 활용될 수 있지만 AA의 이름이나 모델명만으로는 실행 주체의 암호학적 신원을 증명할 수 없다. 본 연구에서는 AA마다 독립적인 개인키와 공개키를 생성하고 CA가 AID, Public Key, Owner, Role, ModelHash, Expiry를 결합한 인증서를 발급한다.

2.3 AI에이전트 신원과 Human-to-Agent 권한위임

최근 Agentic AI의 확산에 따라 AI 에이전트가 누구를 대신하여 어떤 권한 범위에서 동작하는지를 검증하는 신원 및 권한 위임 문제가 중요하게 다루어지고 있다. South 등은 AI 에이전트에 대한 인증되고 감사 가능한 권한 위임(authenticated delegation)의 필요성을 제시하고 OAuth 2.0과 OpenID Connect에 에이전트 전용 자격정보와 권한 범위를 결합하는 방향을 논의하였다[16].

OpenID Foundation의 Agentic AI 신원관리 연구는 에이전트 인증, 권한 부여, 위임 권한의 전파 및 감사 가능성을 핵심 과제로 제시한다[17]. 또한 Benjamin 등은 바이오메트릭 정보와 AI 에이전트 식별자 및 권한 범위를 암호학적으로 결합하여 Human-to-Agent delegation을 검증하는 BIND 프레임워크를 제안하였다[18].

이들 연구는 에이전트의 신원과 권한 위임을 중심으로 하는 반면, 본 연구는 서로 다른 방식으로 인증된 HP와 AA를 동일한 MP 서명 인터페이스로 통합하고 특정 회의, 의사결정, 동적 그룹 상태에 공동서명을 바인딩하여 사후 검증 가능성을 제공한다.

2.4 블록체인 기반 그룹 상태 감사

블록체인은 거래와 상태정보를 순서대로 기록하고, 이전 상태의 해시값을 후속 기록과 연결하여 기록의 변경 여부를 검증하도록 한다[19, 20, 21].

Merkle tree는 다수의 데이터 항목을 하나의 root hash로 요약하여 특정 항목의 포함 여부를 효율적으로 검증할 수 있도록 한다[22].

본 연구에서는 BC를 그룹 상태의 변경 이력과 공개키 버전을 감사하기 위한 용도로 사용한다. HP와 AA의 구성원 목록은 각각 Merkle root HR_v와 AR_v로 요약하고, BC에는 MID, DecID, v, HR_v, AR_v, GPK_v, GS_v 를 기록한다.

구성원 목록과 회의 원문은 별도의 접근통제 저장소에서 관리한다. 블록체인 기반 분산 신원 연구에서도 공개키와 상태정보의 검증 가능성 및 변경 이력 관리가 주요 기능으로 논의되고 있다[23].

2.5 기존 연구와 제안한 방법의 차이

기존 그룹서명 연구는 익명성과 추적성을 중심으로 하고, Schnorr 다중서명 연구는 복수 공개키 소유자의 공동 확인을 하나의 서명으로 결합하는 데 초점을 둔다. 최근 Agentic AI 신원 연구는 에이전트 인증과 권한 위임을 주로 다룬다. 본 연구는 이와 달리 HP와 AA가 동시에 존재하고 두 종류의 참여자가 회의 중 동적으로 가입 및 탈퇴하는 환경을 대상으로 하며, 이종 인증과 동일 서명 인터페이스의 결합, Human-Agent별 상태 구분, 공동 공개키의 동적 갱신, BC 기반 상태 버전 관리와 의사결정 기록 공동서명을 하나의 절차로 통합한다.

3. 제안한 방법

3.1 시스템 구성

제안한 시스템은 HP, AA, MP, CA, MM, BC 및 V(Verifier)로 구성된다. HP는 실제 사람 회의 참여자이며, AA는 회의 요약, 검색, 분석, 의사결정 지원 등의 역할을 수행하는 AI 에이전트이다. 인증 및 가입이 완료된 HP와 AA를 통칭하여 MP라 한다.

$$MP \in \{HP,\ AA\}$$

MM은 회의를 생성하고 MP의 가입, 탈퇴 및 서명 세션을 관리한다. CA는 HP 또는 AA의 공개키와 신원정보를 인증하며, BC는 공동 공개키 및 그룹 상태를 기록한다. V는 공동서명의 유효성을 검증한다.

제안한 방법에서 p는 모듈러 연산에 사용되는 큰 소수이고, q는 p-1의 약수인 소수이다. G는 위수가 q인 순환부분군이며, g는 G의 생성자이다. 모든 개인키와 일회용 난수는 Z_q에서 선택하고 공개키는 G의 원소로 계산한다. H는 암호학적 해시 함수이고, KDF는 키 유도 함수이다.

$$G = \langle g \rangle,\quad |G| = q,\quad q \mid (p-1),\quad g^q \equiv 1 \bmod p$$

3.2 등록과 가입

(1) HP의 등록과 가입

FE(Fuzzy Extractor)는 측정할 때마다 일부 값이 달라질 수 있는 바이오메트릭 정보로부터 암호학적으로 사용할 수 있는 안정적인 비밀값을 생성하는 기법이다[6]. HP_i는 바이오메트릭 정보 B_i를 측정하고 FE를 이용하여 ($seed_i$, $helper_i$)를 생성한다. 재인증에서는 새롭게 측정한 B_i'와 $helper_i$를 이용하여 동일한 $seed_i$를 복구한다.

개인키는 바이오메트릭으로부터 얻은 $seed_i$, 등록 시 단말에서 한 번 생성하여 안전하게 저장한 난수 $rand_i$ 및 GID를 함께 KDF에 입력하여 생성한다.

$$hsk_i = KDF(seed_i \| \mathrm{rand}_i \| GID) \bmod q \quad (1)$$

$$hpk_i \equiv g^{hsk_i} \bmod p \quad (2)$$

HP_i는 hpk_i와 함께 PoP_i를 제출한다. MM은 인증서 유효성, 회의 참여 권한, 인증서 공개키 일치 여부, PoP 및 인증서 폐기 여부를 확인한 후 HP_i의 가입을 승인한다. 원본 바이오메트릭 정보, seed 및 개인키는 BC나 회의 서버에 저장하지 않는다.

(2) AA의 등록과 가입

AA_j는 안전한 난수 생성기를 이용하여 개인키를 생성한다.

$$ask_j \leftarrow Z_q \quad (3)$$

$$apk_j \equiv g^{ask_j} \bmod p \quad (4)$$

CA는 AA_j의 식별자, 공개키, 소유자, 역할, 모델 정보를 확인한 후 다음과 같이 인증서를 발급한다.

$$aCert_j = Sign_{CA}(AID_j \| apk_j \| Owner_j \| Role_j \| ModelHash_j \| \mathrm{Expiry}_j) \quad (5)$$

MM은 CA 전자서명, 인증서 유효기간과 폐기 상태, 역할 권한, PoP 및 ModelHash를 검증한 후 AA_j의 가입을 승인한다.

(3) MP 통합

가입이 완료된 이후 HP와 AA는 동일한 MP로 통합한다. MP의 개인키와 공개키는 다음과 같이 정의한다.

$$msk_l = \begin{cases} hsk_i, & MP_l = HP_i \\ ask_j, & MP_l = AA_j \end{cases}$$

$$mpk_l = \begin{cases} hpk_i, & MP_l = HP_i \\ apk_j, & MP_l = AA_j \end{cases}$$

$$mpk_l \equiv g^{msk_l} \bmod p \quad (6)$$

공동서명 단계에서는 $SignerList_v$에 포함된 MP를 MP_i, i=1,...,k로 재색인한다.

3.3 그룹 상태 구성 및 공동서명 생성

특정 의사결정 DecID에 대한 공동서명이 시작되면 MM은 회의 규칙에 따라 서명 책임을 갖는 MP의 집합인 $SignerList_v$를 확정한다. $SignerList_v$는 전체 회의 참석자 목록과 반드시 동일하지 않으며, 본 논문의 공동서명은 의사결정 결과에 대한 찬성 여부가 아니라 최종 의사

www.kci.go.kr

결정 기록 M의 정확성과 서명 참여 맥락을 확인하는 것을 의미한다. 따라서 반대 의견을 가진 MP도 자신의 반대 의견이 M에 정확히 기록되었음을 확인하고 서명할 수 있다. 본 기법의 n-of-n 특성은 전체 회의 참석자가 아니라 확정된 $SignerList_v$ 내부에 적용된다.

여기서 n은 $SignerList_v$에 포함된 HP의 수, m은 $SignerList_v$에 포함된 AA의 수이며, k=n+m은 해당 의사결정의 공동서명에 참여하는 전체 MP의 수이다.

$$n = |HP \cap SignerList_v|,\ m = |AA \cap SignerList_v|$$

$$k = n + m$$

공동 공개키 GPK_v는 다음과 같이 생성한다.

$$GPK_v = \prod_{i=1}^{k} mpk_i \mod p \quad (7)$$

그룹상태를 나타내는 GS_v는 단순한 구성원 상태가 아니라 특정 회의 및 의사결정에 바인딩된 그룹 상태값을 의미한다. HP 목록과 AA 목록은 각각 Merkle root HR_v와 AR_v로 요약한다. $SignerList_v$, HP 목록 및 AA 목록은 구성원의 고유 식별자를 기준으로 정렬한 정규화된 순서로 구성한다.

$$GS_v = H(GID \| MID \| DecID \| v \| HR_v \| AR_v \| GPK_v \| GS_{v-1}) \quad (8)$$

동시 또는 재시도 서명 세션을 구분하기 위해 MM은 세션마다 새로운 난수 $rand_{MM}$을 생성하고 고유한 서명 세션 식별자 SID를 계산한다.

$$SID = H(MID \| DecID \| v \| rand_{MM}) \quad (9)$$

각 MP는 각 SID 마다 새로운 난수 r_i를 선택하고 임시 공개값 R_i를 계산한다.

$$R_i \equiv g^{r_i} \mod p \quad (10)$$

다른 참여자의 R_i를 확인한 후 자신의 r_i를 의도적으로 선택하거나 변경하는 공격을 방지하기 위해, 각 MP는 R_i를 공개하기 전에 먼저 커밋먼트 값을 제출한다.

$$CommitR_i = H(R_i \| SID) \quad (11)$$

모든 $CommitR_i$가 수집된 후 R_i를 공개하고 MM은 커밋먼트 일치 여부를 검증한다. 검증된 모든 R_i는 다음과 같이 결합된다.

$$R \equiv \prod_{i=1}^{k} R_i \mod p \quad (12)$$

$$msg = H(M \| MID \| DecID \| GID \| v \| GS_v \| SID) \quad (13)$$

$$c = H(msg \| R \| GPK_v \| H(SignerList_v)) \mod q \quad (14)$$

$$s_i \equiv r_i + c \cdot msk_i \mod q \quad (15)$$

MM은 각 부분서명 s_i를 수신할 때 다음 식을 이용하여 부분서명의 유효성을 확인한다. 검증에 실패한 경우 해당 세션을 중단한다.

$$g^{s_i} \equiv R_i \cdot mpk_i^c \mod p \quad (16)$$

$$s \equiv \sum_{i=1}^{k} s_i \mod q \quad (17)$$

$$Sig_v = (MID, DecID, GID, v, GS_v, SID, R, s) \quad (18)$$

3.4 공동서명 검증 및 동적 구성원 관리

V는 BC에서 MID, DecID, GID 및 v에 대응하는 GS_v와 GPK_v를 조회한다. 이후 $SignerList_v$, SID를 확인하고 HR_v, AR_v, GPK_v, msg 및 c를 재계산한 후 다음 식을 검증한다.

$$g^s \equiv R \cdot GPK_v^c \mod p \quad (19)$$

새로운 MP가 가입하면 기존 GPK_v에 신규 MP의 공개키 mpk_{new}를 곱하여 공동 공개키를 갱신한다.

$$GPK_{v+1} = GPK_v \cdot mpk_{new} \mod p \quad (20)$$

MP가 탈퇴하면 해당 MP의 공개키의 역원을 기존 GPK_v에 곱하여 제거한다.

$$GPK_{v+1} = GPK_v \cdot mpk_{leave}^{-1} \mod p \quad (21)$$

구성 변경 이후에는 새로운 HR_{v+1} 또는 AR_{v+1}을 생성하고 그룹 상태를 갱신한다.

$$GS_{v+1} = H(GID \| MID \| DecID \| (v+1) \| HR_{v+1} \| AR_{v+1} \| GPK_{v+1} \| GS_v) \quad (22)$$

서명 세션 중 MP의 가입이나 탈퇴가 발생하면 현재 세션을 폐기하고 새로운 SignerList, GPK, GS, SID 및 난수를 이용하여 공동서명을 다시 시작한다. 이는 서로 다른 그룹 상태 또는 동시 세션의 부분서명이 하나의 최종 공동서명에 혼합되는 것을 방지한다.

4. 안전성 및 효율성 분석

4.1 공격자 모델 및 보안 가정

공격자는 네트워크 상의 메시지를 관찰, 변조, 삽입 및 재전송할 수 있고 일부 MP를 장악하여 악성 공개키나 잘못된 부분서명을 제출하거나 여러 공동서명 세션에 동시에 참여할 수 있다고 가정한다. 또한 MM 또는 CA가 손상되는 경우를 별도의 신뢰 침해 시나리오로 고려한다.

다만 공격자는 이산대수 문제를 효율적으로 해결할 수 없고, 안전한 해시 함수와 CA 전자서명을 위조할 수 없

www.kci.go.kr

으며, 손상되지 않은 MP의 개인키와 안전한 저장영역에는 접근할 수 없다고 가정한다. BC에 기록된 데이터는 해당 블록체인의 합의 보안 가정이 유지되는 한 임의 변경이 어렵다고 가정한다.

4.2 공동서명 정확성 및 상태 바인딩

정리 1. (공동서명의 정확성) 승인된 모든 MP가 제안한 방법에 따라 부분서명을 생성하면 최종 공동서명은 식 (19)를 만족한다.

(증명) 각 부분서명은 $s_i \equiv r_i + c \cdot msk_i$ 이고, $s \equiv \sum_{i=1}^{k} s_i$ 이다. 따라서 $g^s \equiv g^{\sum_{i=1}^{k} r_i} \cdot g^{\sum_{i=1}^{k} c \cdot msk_i}$ 이다. 식 (12)에 의해 $g^{\sum_{i=1}^{k} r_i} = R$ 이고, 식 (7)에 의해 $g^{\sum_{i=1}^{k} msk_i} = GPK_v$이므로 $g^s \equiv R \cdot GPK_v^c \mod p$가 성립한다. Q.E.D.

정리 2. (비인가 공동서명 위조 저항성) 공격자가 $SignerList_v$에 포함된 필수 MP의 개인키를 알지 못하고 사용된 군에서 이산대수 문제를 효율적으로 해결할 수 없다면, 새로운 메시지와 그룹 상태에 대한 유효한 공동서명을 생성하는 것은 계산적으로 어렵다.

(증명) 공격자가 $mpk_i \equiv g^{msk_i}$로부터 msk_i를 계산하려면 이산대수 문제를 해결해야 한다. 또한 challenge c는 M, MID, DecID, GID, v, GS_v, SID, R, GPK_v 및 $SignerList_v$에 의해 결정되므로 과거 부분서명을 새로운 메시지, 다른 그룹 상태 또는 다른 동시 세션에 그대로 사용할 수 없다. 가입 단계에서 PoP를 검증하므로 rogue-key 공격의 위험도 줄어든다. 따라서 Schnorr 서명의 안전성, 이산대수 문제의 어려움 및 CA 서명의 위조 불가능성이 유지되는 한 비인가 공동서명 생성은 계산적으로 어렵다. Q.E.D.

정리 3. (동적 구성의 일관성) 가입 시 새 공개키를 곱하고 탈퇴 시 탈퇴 공개키의 역원을 곱하면, 갱신된 공동공개키는 현재 $SignerList_v$에 포함된 모든 MP의 공개키를 곱한 값과 일치한다.

(증명) 가입 시 $GPK_{v+1} = GPK_v \cdot mpk_{new} \mod p$ 이고, 탈퇴 시 $GPK_{v+1} = GPK_v \cdot mpk_{leave}^{-1} \mod p$ 이며 $mpk_{leave} \cdot mpk_{leave}^{-1} = 1$ 이다. 따라서 항상 $GPK_v \equiv \prod_{MP_i \in SignerList_v} mpk_i$라는 불변식이 유지된다. Q.E.D.

정리 4. (그룹상태 바인딩과 과거서명 재사용 방지) **HP 또는 AA의 가입이나 탈퇴로 그룹 구성 상태가 변경된 경우, 이전 버전에서 생성된 공동서명을 새로운 버전의 유효한 공동서명으로 재사용하는 것은 계산적으로 어렵다.**

(증명) 구성원이 변경되면 HR_v, AR_v, GPK_v 및 v 중 하나 이상이 달라지고, 식 (8)에 의해 GS_v가 변경된다. 이에 따라 msg와 c도 달라진다. 따라서 이전 버전의 공동서명은 새로운 GPK와 c를 사용하는 검증식을 만족하지 못한다. 또한 GS_v가 GS_{v-1}을 포함하고 BC에 기록되므로 과거 상태의 임의 변경도 탐지 가능하다. Q.E.D.

4.3 세부 위협 및 키 관리 분석

- Rogue-key 공격: 가입 시 각 MP가 등록 공개키에 대응하는 개인키 보유를 Schnorr 기반 PoP로 증명하므로 다른 참여자의 공개키를 상쇄하도록 조작된 공개키를 등록하는 공격을 완화한다.
- 난수값 재사용: 동일한 r_i를 서로 다른 challenge c와 c'에서 사용하면 개인키가 노출될 수 있다. 따라서 각 SID마다 새로운 난수값을 생성하고 사용 후 즉시 폐기해야 한다.
- 악성 서명자: MM은 각 부분서명에 대해 식 (16)을 검증하므로 잘못된 부분서명을 탐지할 수 있다. 다만 MP가 부분서명 제출을 거부하는 공격은 무결성 위조가 아니라 가용성 저하에 해당하며 해당 세션은 실패 처리한다.
- 동시 세션: SID를 $CommitR_i$와 msg에 포함함으로써 동일 MID, DecID, v에서도 병렬 세션의 난수값과 challenge가 서로 혼용되지 않도록 한다.
- MM 손상: 손상된 MM은 SignerList 조작, 메시지 지연 또는 세션 중단을 시도할 수 있으나 정직한 MP의 개인키 없이 유효한 공동서명을 직접 위조할 수는 없다. 각 MP는 서명 전에 M, $SignerList_v$, GS_v와 SID를 확인해야 하며 BC 기록을 통해 상태 변경을 감사할 수 있다.
- CA 손상: CA 개인키가 손상되면 허위 인증서 발급이 가능하므로 이는 본 기법의 핵심 신뢰 가정을 위반한다. HSM 기반 키 보호, 짧은 인증서 유효기간, 폐기목록, CA 키 교체와 투명한 감사 로그가 운영상 요구된다.
- helper data와 단말 난수 노출: $rand_i$가 노출된 경우 $seed_i$ 없이는 hsk_i를 계산할 수 없지만, $helper_i$

www.kci.go.kr

와 바이오메트릭에 가까운 입력까지 확보되면 위험이 증가하므로 $rand_i$는 단말의 안전한 저장영역에 보호한다.

- Key recovery 및 revocation: 정상 재인증에서는 B_i'와 $helper_i$로 동일한 $seed_i$를 복구하고 저장된 $rand_i$ 및 GID를 이용해 기존 hsk_i를 재생성한다. 단말 분실로 $rand_i$를 복구할 수 없는 경우 동일 키 복구를 시도하지 않고 기존 인증서와 공개키를 폐기한 뒤 새로운 $rand_i$로 재등록한다. $helper_i$의 무결성이 손상된 경우에도 재등록하여 새로운 $helper_i$를 생성한다. 바이오메트릭 자체가 노출된 경우에는 다른 생체특성이나 추가 인증요소를 이용한 재등록이 필요하다.

4.4 정량적 효율성 비교

실행시간은 하드웨어 환경과 구현 라이브러리에 따라 달라질 수 있으므로, 본 연구에서는 동일한 Schnorr 서명 방식을 기준으로 개별 서명과 제안 공동서명의 핵심 서명 크기를 비교한다. 여기서 k는 $SignerList_v$에 포함된 서명자 수를 의미하며, $|p|$와 $|q|$는 각각 군 원소와 스칼라의 비트 길이를 나타낸다.

개별 Schnorr 서명 방식에서는 각 서명자가 독립적으로 서명을 생성하므로 전체 서명 크기는 서명자 수에 비례하는 $k(|p|+|q|)$가 된다. 반면, 제안한 공동서명 방식은 다수 서명자의 서명을 하나로 통합하므로 최종 서명 크기가 $|p|+|q|$로 일정하게 유지된다. 따라서 제안한 방식은 개별 Schnorr 서명 방식과 비교하여 서명 데이터의 크기를 약 $1/k$ 수준으로 줄일 수 있으며, 서명자 수가 증가할수록 데이터 크기 감소 효과가 커진다.

서명자 수 증가에 따른 확장성 측면에서 개별 Schnorr 서명 방식은 서명자 수에 비례하여 전체 서명 크기가 $O(k)$로 증가하는 반면, 제안한 공동서명 방식은 최종 공동서명의 핵심 암호학적 크기를 $O(1)$로 유지한다. 또한 개별 서명 방식은 사후 검증 시 k개의 서명을 각각 검증해야 하지만, 제안한 방식은 부분서명 생성 단계의 유효성 검증 이후 최종 결과에 대해서는 식 (19)의 하나의 공동서명 검증식만 수행한다. 따라서 제안한 방식은 다수의 서명자가 참여하는 환경에서 서명 데이터의 저장 및 전송 부담을 줄이고 최종 검증 절차를 단순화하는 데 유리하다.

종합하면, 제안한 공동서명 방식은 개별 Schnorr 서명 방식과 비교하여 서명자 수가 증가하더라도 최종 서명 크기가 일정하게 유지된다는 장점이 있다. 이는 다수 서명자가 참여하는 환경에서 서명 데이터의 저장 및 전송 효율성을 높이고, 시스템의 확장성을 향상시키는 데 기여할 수 있다. 다만, SignerList, 인증서, Merkle root 등 상태 및 감사 데이터의 저장, 전송 비용은 별도로 고려해야 한다.

5. 결론

본 논문에서는 HP와 AA가 함께 참여하는 다자간 회의 환경에서 특정 시점의 승인된 참여자 집합과 공동 의사결정 기록의 무결성을 검증하기 위한 Human-Agent 동적 그룹 공동서명 기법을 제안하였다. HP는 바이오메트릭 정보와 FE 및 단말 난수를 이용하고, AA는 난수 기반 전용 개인키와 에이전트 인증서를 이용한다. 인증 이후에는 HP와 AA를 모두 MP로 통합하여 동일한 Schnorr 기반 공동서명에 참여하도록 한다.

본 논문의 공동서명은 의사결정 결과에 대한 만장일치 찬성을 의미하지 않고, 최종 의사결정 기록 M의 내용과 각 서명자의 의견이 정확히 기록되었음을 $SignerList_v$가 공동으로 확인하는 것을 의미한다. 따라서 반대 의견 역시 M에 포함하여 검증 가능한 기록으로 남길 수 있다.

제안한 방법은 과거 회의 또는 구성 변경 이전에 생성한 공동서명을 새로운 그룹 상태에서 재사용할 수 없고, 탈퇴한 MP 역시 새로운 공동서명에 참여할 수 없다. 또한 rogue-key, nonce reuse, malicious signer, concurrent session, MM/CA compromise 및 FE 기반 키 관리에 대한 위협과 한계를 분석하였다.

개별 Schnorr 서명을 각각 저장, 검증하는 방식과 비교할 때 k명의 개별 서명은 크기가 $k(|p|+|q|)$로 증가하는 반면 제안한 기법의 최종 공동서명 핵심 크기는 $|p|+|q|$로 일정하며 최종 검증식도 하나로 유지된다. 다만 공동 공개키와 난수 값 집계 비용은 k에 선형적으로 증가한다.

향후 연구과제는 표준 타원곡선 기반 프로토타입에서 MP 수에 따른 가입, 공동 공개키 갱신, 공동서명 생성 및 검증시간과 BC 기록 비용을 효율적으로 수행하고 TEE 기반 AA 실행환경 검증을 결합하여 실용적인 Human-Agent 회의 인증 구조로 확장하는 것이다.

www.kci.go.kr

REFERENCES

[1] Q.Wu et al., "AutoGen: Enabling next-gen LLM applications via multi-agent conversation," Proceedings of the Conference on Language Modeling (COLM), 2024.

[2] G.Chen et al., "AutoAgents: A framework for automatic agent generation," arXiv preprint arXiv:2309.17288, 2023.

[3] J.S.Park et al., "Generative agents: Interactive simulacra of human behavior," Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology (UIST '23), Article 2, pp. 1–22, 2023.

[4] Z.Xi et al., "The rise and potential of large language model based agents: A survey," arXiv preprint arXiv:2309.07864, 2023.

[5] S.Yun, "The Biometric Authentication based Dynamic Group Signature Scheme," Journal of the Korea Convergence Society, vol. 7, no. 1, pp. 49–55, 2016.

[6] Y.Dodis, R.Ostrovsky, L.Reyzin, and A.Smith, "Fuzzy extractors: How to generate strong keys from biometrics and other noisy data," SIAM Journal on Computing, vol. 38, no. 1, pp. 97–139, 2008.

[7] National Institute of Standards and Technology, Digital Identity Guidelines: Authentication and Authenticator Management, NIST SP 800-63B-4, 2025.

[8] D.Chaum and E.van Heyst, "Group signatures," in Advances in Cryptology—EUROCRYPT '91, LNCS 547, pp. 257–265, 1991.

[9] M.Bellare, D.Micciancio, and B.Warinschi, "Foundations of group signatures: Formal definitions, simplified requirements, and a construction based on general assumptions," in Advances in Cryptology—EUROCRYPT 2003, LNCS 2656, pp. 614–629, 2003.

[10] D.Boneh, X.Boyen, and H.Shacham, "Short group signatures," in Advances in Cryptology—CRYPTO 2004, LNCS 3152, pp. 41–55, 2004.

[11] G.Ateniese, J.Camenisch, M.Joye, and G.Tsudik, "A practical and provably secure coalition-resistant group signature scheme," in Advances in Cryptology—CRYPTO 2000, LNCS 1880, pp. 255–270, 2000.

[12] C.P.Schnorr, "Efficient signature generation by smart cards," Journal of Cryptology, vol. 4, no. 3, pp. 161–174, 1991.

[13] G.Maxwell, A.Poelstra, Y.Seurin, and P.Wuille, "Simple Schnorr multi-signatures with applications to Bitcoin," Designs, Codes and Cryptography, vol. 87, pp. 2139–2164, 2019.

[14] J.Nick, T.Ruffing, and Y.Seurin, "MuSig2: Simple two-round Schnorr multi-signatures," in Advances in Cryptology—CRYPTO 2021, LNCS 12825, pp. 189–221, 2021.

[15] W3C, Decentralized Identifiers (DIDs) v1.0: Core Architecture, Data Model, and Representations, W3C Recommendation, 2022.

[16] T.South, S.Marro, T.Hardjono, R.Mahari, C.D.Whitney, A.Chan, and A.Pentland, "Position: AI Agents Need Authenticated Delegation," Proceedings of the 42nd International Conference on Machine Learning, PMLR, vol. 267, pp. 82211–82231, 2025.

[17] T.South et al., "Identity Management for Agentic AI: The New Frontier of Authorization, Authentication, and Security for an AI Agent World," OpenID Foundation Whitepaper, 2025.

[18] J.G.Benjamin, A.K.Jain, and K.Nandakumar, "Binding Biometrics with AI Agent Identifiers for Delegation of Authority," arXiv preprint arXiv:2608.04292, 2026.

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

[20] S.Haber and W.S.Stornetta, "How to time-stamp a digital document," Journal of Cryptology, vol. 3, pp. 99–111, 1991.

[21] D.Yaga, P.Mell, N.Roby, and K.Scarfone, Blockchain Technology Overview, NISTIR 8202, 2018.

[22] R.C.Merkle, "Protocols for public key cryptosystems," in Proceedings of the IEEE Symposium on Security and Privacy, pp. 122–134, 1980.

[23] Z.Yan, X.Zhao, Y.Liu, and X.Luo, "Blockchain-driven decentralized identity management: An interdisciplinary review and research agenda," Information & Management, vol. 61, no. 7, Art. no. 104026, 2024.

윤 성 현(Yun Sunghyun) [종신회원]



- 1997년 2월 : 고려대학교 일반대학원 컴퓨터학과 (이학박사)
- 1998년 3월 ~ 2002년 2월 : LG 전자 선임연구원
- 2002년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수

〈관심분야〉

블록체인, AI, 사이버보안, 사물인터넷, DRM, 정보보호

www.kci.go.kr