

CBDC 시스템 R-ABAC 접근통제 모델 설계 및 검증

김지민¹, 박건우², 김솔리¹

¹서울여자대학교 정보보호학과 학부생

²중앙대학교 산업보안학과 학부생

carrp@swu.ac.kr, pomol4@cau.ac.kr, so1io@swu.ac.kr,

Design and Verification of an R-ABAC Access Control Model for CBDC Systems

Ji-Min Kim¹, Gun-Woo Park², Soli Kim¹

¹Dept. of Information Security, Seoul Women's University

²Dept. of Industrial Security, Chung-Ang University

요 약

본 연구는 중앙은행 디지털화폐(CBDC) 시스템에서 발생할 수 있는 권한 상승 위협에 대응하기 위해 R-ABAC(Role-Attribute Based Access Control) 모델 기반 접근 제어 정책을 설계하고 검증하였다. Alloy Analyzer와 OPA(Open Policy Agent)를 활용한 정적·동적 검증을 통해 정책의 논리적 일관성과 운영 환경 적용 가능성을 확인하였다.

1. 서론

중앙은행 디지털화폐(CBDC)는 민간 스테이블코인의 등장과 코로나 19 이후 확산된 비대면 결제를 계기로 글로벌 금융시장에서 연구와 시범 도입이 점차 확대되고 있다. 한국은행 역시 CBDC 시험 프로젝트 ‘한강’을 추진하였으며, 이 과정에서 발생할 수 있는 보안 위협은 여전히 주요한 연구 대상이다.

DFD를 기반으로 CBDC 시스템의 STRIDE 위협 모델링을 수행한 결과, 120개 위협 중 약 27.5%가 권한 상승(Elevation of Privilege) 범주에 속하여 6개 위협 중 가장 높은 비중을 차지하였다.

이에 대응하기 위해 본 연구에서는 R-ABAC(Role-Attribute Based Access Control) 모델을 설계하고, 정적·동적 검증을 통해 설계 단계의 논리 일관성과 운영 단계의 정책 집행력을 동시에 확보하는 것을 목표로 한다.

2. R-ABAC 모델 설계 및 접근 제어 정책 설계

국제결제은행(BIS)은 CBDC 보안 설계의 핵심 원칙으로 ZTA(Zero Trust Architecture)를 명시하였으며[1], ZTA는 최소 권한 원칙, 지속적 모니터링, 세분화된 접근제어를 요구한다.

이에 따라 미국 사이버 보안 및 인프라보안국(CISA)은 ZTA 모델 구현을 위한 접근 제어 방식으로

RBAC 또는 ABAC 중 하나를 활용할 것을 권고하고 있으며[2], NIST SP 800-162, 개인정보보호법, 금융보안원 가이드 등에서도 접근제어 정책 설계와 도입에 대한 권고사항을 제시하고 있다.

RBAC은 단순하고 관리 효율성이 높지만 역할 수가 많아질수록 표현에 제약이 있고, ABAC은 세밀한 정책 설정이 가능하지만 설계와 운영 측면에서 부담이 크다.

이를 CBDC 권한 관리 체계에 적용하여 두 모델의 한계를 상호 보완한 보안 정책을 제시하였다.

구성요소	정의	특징	정책 표현 예시
1 사용자 속성 (UATT)	사용자에 대한 속성 정보	- MFA 여부, 소속(은행 직원, 일반 사용자, 규제기관 등) 등을 포함	MFA(s) = True,
2 객체 속성 (OATT)	접근 대상 객체에 대한 속성 정보	- owner_id, object_type 등을 포함	object.type = Account
3 속성 타입 (attType)	속성 값의 데이터 유형	- (atomic, set) 중 하나로 명시 - 값을 명시하여 할당	owner_id = 123 verificationDevice = session.device
4 권한 필터링 정책 (PFP)	UATT와 OATT를 조합한 필터링 규칙	- 단순한 역할 할당을 넘어 조건부 접근 제어 가능 - 필터링 언어로 작성	- 금액 금액(amount)이 1천만 이상일 시, MFA 인증 필수 - Account 객체 접근은 owner가 사용자 할당 시 허용 - Issue 연산은 CB_Admin인 Currency에 가능
5 필터 함수 (FILTER)	주어진 세션, 자원, 객체에 대해 권한 여부를 반환하는 함수	- Boolean 함수로 T, F를 반환	F(session, operation, object) → (T, F)
6 대상 필터 (TargetFilter)	객체의 속성에 따라 어떤 필터 함수를 적용할지 매핑	- 객체 종류마다 다른 규칙을 적용	(op = View → P1[s,o] or P7[o,o])
7 필터링 언어 (Filter, LCondition)	정책 표현에 사용되는 CPL 기반 논리 언어	- 집합 포함($\subseteq$, $\in$), 원자 비교($=$, $<$, $>$), 논리 연산($\wedge$, $\vee$, $\neg$) 지원	balance(s) > transactionAmount $\wedge$ MFA(s) = True

<표 1> R-ABAC 모델 구성요소와 정책 표현 예시

R-ABAC 구성요소를 기반으로, 정책 설계는 다음과 같이 이루어졌다.

첫째, 사용자 속성(UATT)과 객체 속성(OATT)을 중심으로 기본 속성을 구성하였다. 예를 들어, 사용자의 MFA 인증 여부, 소속, 객체의 소유자 정보(owner_id),

객체 유형(Account 등)에 따라 시스템 접근 권한이 세분화된다.

둘째, 속성 타입(attType) 및 필터 함수(FILTER)를 통해 속성의 데이터 타입과 조건 검사를 체계화하였으며, 필터 정책(PFP)과 대상 필터(TargetFilter)를 조합하여 상황별 정책 적용을 설계하였다.

셋째, 정책 논리(Filter, LCondition)를 고려하여 정책 조건을 논리식으로 결합하고, 우선순위를 제어할 수 있도록 하였다.

번호	정책 설명	역할	작업 행위	객체/속성
1	사용자는 자신의 계좌에 조회 가능	User	view	본인 계좌
2	사용자는 자신의 계좌에 결제 요청 가능	User	pay	본인 계좌
3	사용자는 고객 결제 시 MFA 필수	User	pay	본인 계좌
4	사용자는 고객 결제 관련 요청은 등록 가능	User	update, patch	한도설정 시스템 서버
5	시스템 관리자는 시스템 유지보수할 가능	Sys.Operator	update, patch	시스템 서버
6	Regulator는 로그 조회 가능	Regulator	view	로그 DB
7	사용권한 제한은 운영시간에 의해서만 가능	Bank_User	view	고객 정보 서버
8	사용권한 제한은 운영시간에 의해서만 가능	Bank_User	register	고객 정보 서버
9	한국은행 관리자는 정책 설정 가능	CB_Admin	issue, set_policy	정책 서버
10	한국은행 관리자는 정책 설정 가능	CB_Admin	set	한도설정 시스템 서버
11	한국은행 관리자는 정책 설정 가능	User	view	시스템 관리 시스템 서버

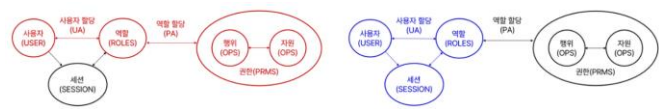
<표 2> 접근 제어 정책 목록

접근 제어 정책은 명시적 허용 정책과 명시적 거부 정책을 병행 사용하여, 허용 정책 만으로는 통제하기 어려운 세밀한 제어가 가능하도록 하였다.

3. 정책 검증 설계

정책 검증은 Alloy Analyzer와 OPA를 병행 활용하여 설계 단계에서는 정책 논리의 타당성을, 운영 환경에서는 정책 실행 제어를 효과적으로 보장하는지 검증하고자 했다.

R-ABAC 모델 구조를 기준으로, Alloy Analyzer를 활용한 정적 검증은 사용자·역할·권한 간의 정적인 관계를 모델링하여 정책의 논리적 일관성을 확인하고, OPA를 활용한 동적 검증은 일부 역할만 실제 세션에서 활성화되는 상황을 가정하여 정책이 운영 환경에서도 의도대로 작동하는지 입증하였다.



(그림 1) Alloy Analyzer와 OPA의 검증 영역

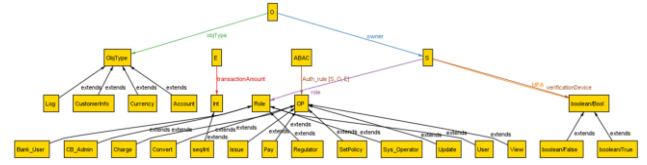
따라서 두 검증에서 논리적 결함이 발견되지 않는 경우, 정책 모델은 구조적으로 안정적이며 실행 환경에서도 충분한 실현 가능한 것으로 판단할 수 있다.

그림 1에서는 NIST의 R-ABAC 모델 구조[3]에서 일부 단순화한 구조를 차용하였다.

4. 정책 검증 결과

Alloy Analyzer 실행 결과, 모든 assert문이 위배되

지 않아 정책에 논리적 모순이 없음을 확인하였다. 서로 다른 접근 정책들은 PFP를 통해 효과적으로 통합되어 충돌이 발생하지 않음을 검증하였다. 또한, 관계 집합의 인스턴스를 실제로 생성함으로써 정책이 실제 시스템에도 적용 가능함을 확인하였다.



(그림 2) Alloy Analyzer 실행 결과 시각화 자료

그림 1은 주체(S), 객체(O), 환경(E)의 세 가지 요소가 명확히 구분되어 시스템에 반영되었으며, Auth_rule[S, O, E] 형태로 통합되어 다양한 접근 정책이 동시에 적용될 수 있음을 보여준다.

번호	테스트 시나리오	예상 결과	관한상승 종류
1	Bank_User가 근무시간에 소속 은행 거래 정보 조회	허용	수평적
2	MFA 없는 User가 고객 송금	거부	수평적
3	Bank_User가 근무시간에 외부에서 타 기관 계좌 조회	거부	수평적
4	시스템 관리자가 정책 서버 수정	거부	수직적
5	19세 이상의 MFA 있는 User가 고객 송금	허용	-
6	User가 다른 User 계좌 조회	거부	수평적
7	User가 본인 계좌 조회	허용	-
8	User가 Bank_User 권한 승급 승인	거부	수직적

<표 3> OPA 테스트 시나리오 15가지

OPA 실행 결과, 수평적·수직적 두 가지 관점에서 총 15개의 임의 권한 상승 시나리오에 대한 정책의 유효성과 정확성을 확인하였다.

5. 결론 및 향후 연구 방향

본 연구는 CBDC 환경에서 발생할 수 있는 권한 상승 위협에 대응하기 위해 R-ABAC 정책 기반 보호 체계를 설계·구현하였다. 이 모델은 실제 CBDC 시스템의 보안 아키텍처와 정책 수립에 활용 가능하며, 향후 모의 환경에서 조직 규모와 보안 요구 수준에 따라 유연하게 적용 가능하도록 추가 검증할 계획이다.

참고문헌

- [1] BIS, Project Polaris : A security and resilience framework for CBDC systems, p. 23, 2023.
- [2] CISA, Compendium of Cybersecurity and Infrastructure Security Agency (CISA) Technology Evaluations, p. 46, 2023.
- [3] X. Jin, R. Sandhu, R. Krishnan, RABAC: Role-Centric Attribute-Based Access Control, Proceedings of the 6th International Conference on Mathematical Methods, Models and Architectures for Computer Network Security (MMM-ACNS 2012), 2012, pp. 84-96.