November 01, 2024

CS61065 Theory and Applications of Blockchain

Assignment 5: Hyperledger Indy

Date of Submission: November 14, 2024 EOD

Note that this is a group submission. Only one member of each group should submit the assignment. Clearly mention your group details in the submission.

Arjun is an environmental researcher who has developed a novel method for sustainable agriculture. He wants to present his research at the International Environmental Summit (IES). To be eligible, he needs to submit proof of his research's authenticity and proof of his academic qualifications. The Summit organisers require verification of these credentials before allowing him to present.

In this assignment, you need to implement a verifiable credential and verifiable presentation flow using Hyperledger Indy, involving the following four parties:

- **University** (Issuing the academic credentials)
- **Arjun** (The researcher)
- **Research Institute** (Issuing the research authenticity credentials)
- **IES Organizers** (Verifying the credentials)

Assume the University will issue credentials for proof of academic qualifications, and the Research Institute will issue credentials for the authenticity of the research. Arjun will then present these credentials to the IES Organizers, who will validate his claims.

Part A (15 Marks):

Launch the Indy pool by starting the Docker image (Check here)

1. Connect to the Indy pool.
2. Configure one steward.
3. Register Verinyms for Trust Anchors - University and Research Institute.

Part B (15 Marks):

1. Set up the credential schemas and credential definitions for AcademicQualification and ResearchAuthorship. The University creates the schema for AcademicQualification and Research Institute creates schema for ResearchAuthorship on the Indy ledger.
2. The Research Institute registers a credential definition for ResearchAuthorship, and the University registers a credential definition for AcademicQualification.

The schema for AcademicQualification and ResearchAuthorship are as follows:

{
"name": "AcademicQualification",
"version": "1.5",
"attributes": [“student_first_name”, ”student_last_name”, "degree", "field_of_study", "university_name", "graduation_year", "cgpa"]
}
{
"name": "ResearchAuthorship",
"version": "1.5",
"attributes": [“author_first_name”, “author_last_name”, "research_title", "institute_name", "research_field", "publication_year"]
}

Part C (10 Marks):

Once the schema and credential definition setup is done, the issuers issue credentials to Arjun:

1. The University issues the ‘AcademicQualification’ credential.
2. The Research Institute issues the ‘ResearchAuthorship’ credential.
3. Arjun saves both credentials to his wallet.

Use the following claims to create the verifiable credentials:
First Name: “Arjun”
Last Name: “Verma”
Degree: “PhD in Environmental Science”
Field of Study: “Sustainable Agriculture”
University Name: “Delhi University”
Graduation Year: 2022
CGPA: 9

Research Title: “Innovative Techniques in Sustainable Farming”
Institute Name: "GreenEarth Research Institute"
Research Field: "Environmental Studies"
Publication Year: 2023

Part D (10 Marks):

The IES Organizers request a “presentation_eligibility_proof_request”, where the proofs for the following are required:

- first_name
- last_name
- degree
- field_of_study
- university_name
- graduation_year [>= 2020 and <= 2023]
- cgpa [> 6]
- research_title
- publication_year [>=2022]
- research_field

The claims in blue must be from a credential issued by the University.

Claims in red must be from a credential issued by the Research Institute.

For graduation_year, cgpa, and publication_year, the values are not requested directly; instead, zero-knowledge proofs are used to validate them (use ‘requested_predicates’ for this).

Submission Instructions

Create a directory and name it as A5_Indy_(#ROLLNUMBERs separated by underscore). You need to create a single file which will execute the entire flow, and place it inside this directory. If you are implementing it in python, then name the file as indyassignment.py. Similarly for nodejs, name it as indyassignment.js, and so on. In the first line of the file, write your roll number as a comment. Compress the folder as a zip (with .zip extension). Upload the compressed file in moodle. Make sure you DO NOT include node_modules or similar library dependency files in your zip.