1. System Overview

Our UAV authentication system uses a Proof of History (PoH) consensus mechanism within a permissioned consortium blockchain. Unlike traditional blockchains that require heavy computational work, PoH creates a chronological record by sequential hashing, making it ideal for resource-constrained UAVs.

Key Components:

PoH Blockchain Core: Maintains the verifiable sequence of events

Consortium Network: Permissioned nodes that validate authentication

ECC Cryptography Module: Handles secure key exchange and encryption

UAV Client: Lightweight implementation for UAV integration

Authentication Protocol: Defines how UAVs authenticate to the network  
  
  
  
5. Explaining the System to Your Professor

When explaining the system to your professor, emphasize these key aspects:

5.1. Proof of History Implementation

"My implementation uses a Proof of History consensus mechanism, which creates a verifiable chronological record of events without requiring intensive computation. Each 'tick' in the PoH sequence contains a hash derived from the previous hash plus any new data. This creates a cryptographically verifiable timeline that cannot be altered without detection."

5.2. Lightweight Design for UAVs

"The system is specifically designed for resource-constrained UAVs. Unlike traditional blockchains that use Proof of Work and require significant computational power, our PoH approach allows even lightweight UAVs to participate in secure authentication with minimal battery drain."

5.3. Security Features

"The system implements multiple security features:

Elliptic Curve Cryptography: Each UAV uses Ed25519 keys for secure, efficient signatures

Nonce Verification: Prevents replay attacks by requiring a unique nonce for each authentication

Immutable History: All authentication events are permanently recorded in the blockchain

Chronological Proof: The PoH mechanism ensures events can be verified in their correct sequence"

5.4. Consortium Model

"Our system uses a permissioned consortium blockchain model where trusted entities (like aviation authorities, manufacturers, and operators) can participate in transaction validation. This maintains security while allowing for governance over who can access the network."

6. Demonstration Script

Here's a step-by-step demonstration script you can follow:

Start the server and explain: "This server implements our consortium node with the Proof of History blockchain."

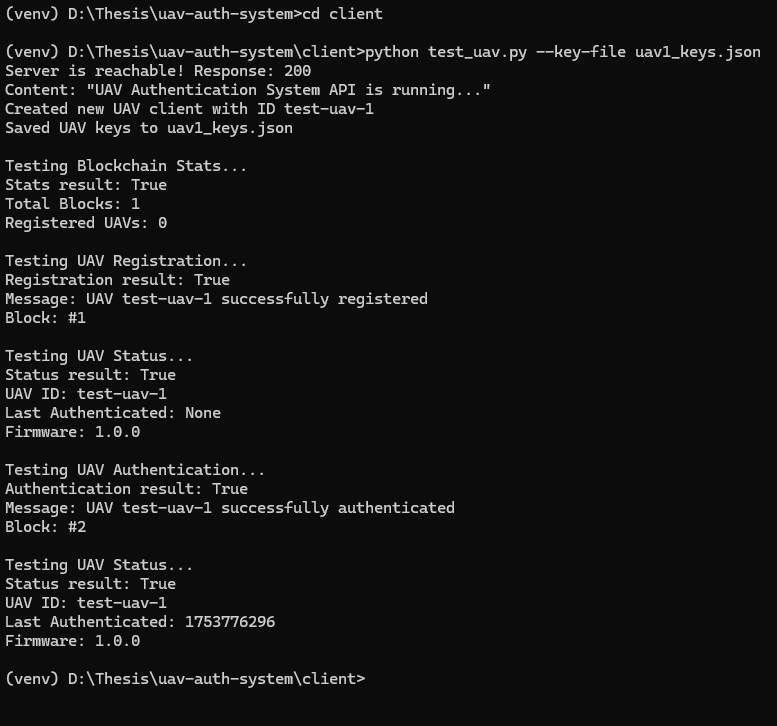
Show blockchain initialization: "When the server starts, it creates a genesis block and initializes the PoH sequence."

Register a UAV: "Now we'll register a UAV with our system. This creates a unique identity for the UAV on the blockchain."

Show blockchain growth: "Notice that registering the UAV created a new block. Each block contains a PoH tick that cryptographically proves when the event occurred."

Authenticate the UAV: "Before a UAV can fly, it must authenticate with the system. The UAV creates a unique nonce and signs it with its private key."

Demonstrate replay attack prevention: "If we try to use the same authentication message twice, the system rejects it, preventing replay attacks."

Show blockchain immutability: "All of these events are permanently recorded in the blockchain. If someone tries to modify a past authentication, the hash chain would break."  
  
API Documentation with Swagger/OpenAPI  
  
<http://localhost:8080/docs>  
  


## 4. Running the System

**4.1. Start the Server**

PowerShell

# Activate the virtual environment if not already activated

cd D:\Thesis\uav-auth-system

.\venv\Scripts\activate

# Start the server

cd server

python server.py

**4.2. Run the Client Tests (in a new PowerShell window)**

PowerShell

# Activate the virtual environment in the new window

cd D:\Thesis\uav-auth-system

.\venv\Scripts\activate

# Run the test client

cd client

python test\_uav.py --key-file uav1\_keys.json