Phạm Quang Duy

0902431405 | duypham140504@gmail.com | LinkedIn | GitHub

SUMMARY

I am a third-year IT student with a strong foundation in Cryptography and Backend Development. Familiar with encryption standards (AES, RSA), secure database practices (e.g., hashing, access control), and network security techniques (firewalls, SSL/TLS, MITM protection).
Passionate about blockchain security. Recently focused on analyzing smart contract vulnerabilities, developing secure Solidity applications, and simulating on-chain attacks and mitigations. Seeking opportunities to grow and learn in the field of Web3, especially through hands-on experience in secure development and smart contract auditing.

EDUCATION

University of Science - VNUHCM **2022-2026**
Third-year student in Bachelor of Information Technology
Minors: Knowledge Engineering - Cryptography & Security

Achivements

- Current GPA (8 semesters): 8.92 / 10
- University Semester Scholarship – University of Science – Semester 2 2024 (9.77), Semester 3 2024 (9.20)

SKILLS

Programming Languages

- **Primary:** Python, Solidity, C/C++
- **Secondary:** HTML, JavaScript, TypeScript (used with Web3.js / Ethers.js for smart contract interaction)

Blockchain & Smart Contract Security

- Smart Contract Development: Solidity, Hardhat, Truffle, Ganache, Remix IDE
- On-chain transaction analysis with Ethers.js / Web3.js, MetaMask, and testnets

Security & Cryptography

- Applied cryptography: AES, RSA, Hash functions (SHA, HMAC, PBKDF2)
- Understanding of SSL/TLS, MITM protection, basic firewall & Kerberos principles

Backend & Tools

- Flask (REST APIs), PostgreSQL / MySQL with secure practices (access control, hashing)
- Git, GitHub, VSCode, VMWare; LaTeX for technical writing
- IPFS tools: Pinata, Web3.Storage

Soft Skills

- Problem-Solving, Technical Writing
- Self-Learning, Collaboration, Time Management

Languages

- English – proficient in reading docs, writing technical reports, and verbal communication

PROJECTS

Blockchain Projects

Blockchain Lottery DApp with NFT (May 2025 – Finished) **[GitHub]**

- Developed a decentralized lottery game using Solidity, Truffle, and Ganache, rewarding winners with unique NFTs.
- Built and tested smart contracts (LotteryMechanism, RandomGenerator, SafeTransfer) with NFT metadata hosted on GitHub Pages.
- Designed user-friendly frontend (HTML/CSS/JS) integrated with Web3.js / Ethers.js to interact with the blockchain.
- Key features:

  - Join lottery rounds with fixed ETH cost per entry.
  - Randomly select winner using internal RandomGenerator contract.
  - Automatically mint and assign NFT rewards to winners.
  - Track players and winners per round; highlight winners with gold styling.
  - Admin-only reset function to start new rounds.

NFT Marketplace Mini (June 2025 – Present) **[GitHub]**

- Built a lightweight NFT marketplace simulating OpenSea using Solidity and Hardhat.
- Implemented ERC721 minting with metadata stored on IPFS (via Pinata).
- Developed smart contracts for NFT minting (‘NFTCollection.sol‘) and trading (‘Marketplace.sol‘) with full on-chain logic.
- Created a minimal frontend using TypeScripts + Ethers.js integrated with MetaMask for user interactions.
- Key features:
  - Mint NFT with custom metadata (name, image, description) hosted on IPFS.
  - List NFTs for sale by approving marketplace and setting price.
  - Buy NFTs using ETH; ownership is transferred via ‘transferFrom‘.
  - View owned NFTs and their metadata directly from IPFS.

Demo Blockchain Website (May 2025 – Finished) **[GitHub]**

- Simulated blockchain processes: mining, block creation, hashing visualization.
- Built using Flask, MySQL, HTML/CSS, and JavaScript.

Security & Cryptography Projects

Homomorphic Voting System (May 2025 – Finished) **[GitHub]**

- Built a secure e-voting platform using Paillier Homomorphic Encryption with Flask + JS frontend.
- Ensures vote confidentiality: encrypted on frontend, only decrypted at tallying stage.
- Implemented:
  - Encrypted vote submission via public key (Paillier).
  - Homomorphic aggregation of votes without decryption.
  - Persistent encrypted vote storage (JSON) across restarts.
  - CORS-enabled API with Flask backend and JS frontend integration.

File/Folder Locker – Python GUI (April 2025 – Finished) **[GitHub]**

- Developed a multi-user file/folder encryption tool with GUI using Tkinter.
- Features:
  - File/folder encryption & decryption (AES-GCM, PBKDF2-HMAC-SHA256)
  - Master password, password change, auto-backup after encryption
  - Supports large files via streaming, user-friendly GUI

Student Management App (March 2025 – Finished) **[GitHub]**

- Built with C# to manage students and teachers with user accounts.
- Implemented features: login for teachers, add/update students, encrypted data transmission with hashing and symmetric encryption.