

# BÁO CÁO ĐỒ ÁN AN NINH MÁY TÍNH 1

## Hệ thống Bảo mật với Mã hóa và Chữ ký Số

### THÔNG TIN NHÓM

Thành viên	MSSV	Email	Vai trò
Phan Thế Anh	22127021	ptanh22@clc.fitus.edu.vn	Cập nhật tài khoản, Khôi phục tài khoản, Báo cáo, Video demo
Nguyễn Khánh Hoàng	22127127	nkhoang22@clc.fitus.edu.vn	Quản lý khoá RSA (cá nhân và các public key), QR code, Chữ ký số và xác minh chữ ký
Lê Thanh Minh Trí	22127422	ltmtri22@clc.fitus.edu.vn	Đăng ký, Đăng nhập/MFA, Giới hạn đăng nhập, Phân quyền tài khoản, Tìm kiếm public key, Mã hóa và giải mã tập tin, Tùy chọn định dạng lưu, Log bảo mật, Quản lý dữ liệu

## 1. TỔNG QUAN DỰ ÁN

### 1.1 Mục tiêu

Xây dựng hệ thống bảo mật desktop hoàn chỉnh với các tính năng:

- Đăng ký, xác thực người dùng với MFA
- Quản lý khoá RSA với thời hạn 90 ngày
- Cho phép xuất và nạp QR Code cho public key
- Tìm kiếm public key theo email và xem danh sách các public key đã lưu
- Mã hóa/giải mã tập tin hybrid (AES + RSA)
- Chữ ký số và xác minh tài liệu
- Phân quyền quản trị và audit logging
- Cập nhật thông tin tài khoản (profile, passphrase)
- Recover tài khoản với mã khôi phục (cho phép đặt lại passphrase và MFA)

### 1.2 Ngôn ngữ & Framework

- Ngôn ngữ:** Python 3.12+
- GUI Framework:** Tkinter
- Cơ sở dữ liệu:** SQLite3
- Thư viện mã hóa:** pycryptodome, cryptography
- Xác thực MFA:** pyotp (TOTP)
- QR Code:** qrcode, pyzbar

## 1.3 Cấu trúc thư mục

```

ComputerSecurityProject/
├── main.py
├── requirements.txt
└── gui/
    ├── main_window.py
    ├── login_frame.py
    ├── register_frame.py
    ├── key_management_frame.py
    ├── encrypt_frame.py
    ├── decrypt_frame.py
    ├── signature_frame.py
    ├── verify_frame.py
    ├── admin_dashboard.py
    └── ...
modules/
└── core/
    └── session.py
└── utils/
    ├── db_helper.py
    ├── crypto_helper.py
    ├── rsa_key_helper.py
    ├── file_crypto_helper.py
    ├── signature_helper.py
    ├── otp_helper.py
    └── logger.py
data/
└── users.db
└── keys/
    └── logs/

```

# Điểm khởi động ứng dụng  
# Dependencies  
# Giao diện Tkinter  
# Cửa sổ chính  
# Giao diện đăng nhập  
# Đăng ký tài khoản  
# Quản lý khóa RSA  
# Mã hóa tệp tin  
# Giải mã tệp tin  
# Tạo chữ ký số  
# Xác minh chữ ký  
# Quản trị hệ thống

# Quản lý phiên đăng nhập

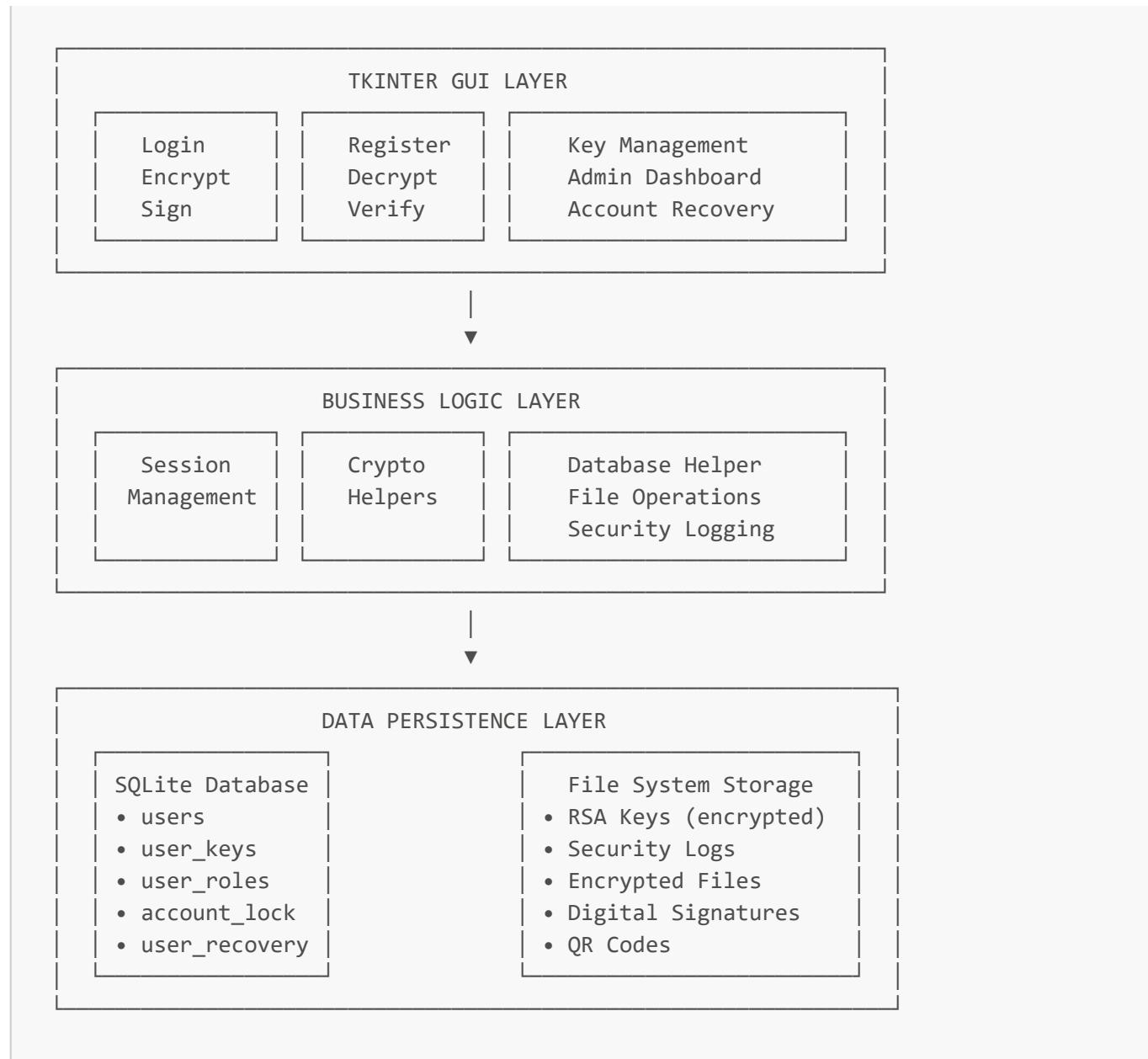
# Thao tác cơ sở dữ liệu  
# Hashing mật khẩu  
# Quản lý khóa RSA  
# Mã hóa/giải mã tệp tin  
# Chữ ký số  
# TOTP/MFA  
# Ghi log bảo mật

# SQLite database  
# Lưu trữ khóa RSA  
# Log bảo mật

## 2. KIẾN TRÚC HỆ THỐNG

### 2.1 Sơ đồ kiến trúc tổng quan





## 2.2 Luồng xử lý chính

### Registration Flow

Đăng ký → Kiểm tra email trùng lặp → Thiết lập TOTP → Mã hóa passphrase → Lưu thông tin vào database → Hiển thị recovery code → Tạo khóa RSA

### Authentication Flow

Đăng nhập → Kiểm tra email/passphrase → Xác thực TOTP → Thiết lập session

### Key Creation Flow

Tạo khóa RSA → Tạo AES session key từ passphrase → Mã hóa private key bằng AES → Lưu trữ an toàn

## Key Management Flow

Xem thông tin khóa → Kiểm tra hết hạn khóa → Gia hạn khóa RSA → Xuất public key dưới dạng PEM/QR Code (và xuất private key dạng PEM nếu cần)

## Public Key Management Flow

Tìm kiếm public key theo email hoặc nạp QR Code → Hiển thị thông tin khóa → Tải public key và metadata về tài khoản → Xem danh sách các public key đã lưu → Tải lại public key mới nếu bản đang có hết hạn

## File Encryption Flow

Chọn file → Sinh AES session key → Mã hóa file → Mã hóa AES key bằng RSA → Lưu kết quả

Note: Public key của người nhận phải đang còn hiệu lực.

## Digital Signature Flow

Chọn file → Tính hash SHA-256 → Ký bằng RSA private key → Lưu chữ ký

Note: Public key của người nhận phải đang còn hiệu lực.

## Account Recovery Flow

Nhập recovery code → Kiểm tra hash → Đặt lại passphrase mới (và MFA nếu muốn) → Mã hóa lại private key → Lưu thông tin mới

# 3. THIẾT KẾ CƠ SỞ DỮ LIỆU

## 3.1 Schema SQLite

```
-- Bảng người dùng chính
CREATE TABLE users (
    id INTEGER PRIMARY KEY AUTOINCREMENT,
    email TEXT UNIQUE,
    full_name TEXT,
    dob TEXT,
    phone TEXT,
    address TEXT,
    passphrase_hash TEXT,          -- SHA-256 hash
    salt TEXT,                     -- Random salt
    totp_secret TEXT,             -- TOTP secret key
    fail_count INTEGER DEFAULT 0, -- Đếm lần đăng nhập sai
    lock_until INTEGER DEFAULT NULL -- Thời gian khóa tài khoản
);

-- Bảng quản lý khóa RSA
CREATE TABLE user_keys (
    email TEXT UNIQUE,
    created_at INTEGER,           -- Timestamp tạo khóa
    expire_at INTEGER,            -- Timestamp hết hạn (90 ngày)
    FOREIGN KEY (email) REFERENCES users(email)
);

-- Bảng phân quyền
CREATE TABLE user_roles (
    email TEXT UNIQUE,
    role TEXT NOT NULL CHECK(role IN ('user', 'admin')),
    FOREIGN KEY (email) REFERENCES users(email)
);

-- Bảng khóa tài khoản
CREATE TABLE account_lock (
    email TEXT UNIQUE,
    locked INTEGER NOT NULL CHECK(locked IN (0, 1)),
    FOREIGN KEY (email) REFERENCES users(email)
);

-- Bảng khôi phục tài khoản
CREATE TABLE user_recovery (
    email TEXT UNIQUE,
    recovery_code_hash TEXT,
    created_at INTEGER,
    FOREIGN KEY (email) REFERENCES users(email)
);
```

### 3.2 Cấu trúc file hệ thống

```
data/
└── users.db                      # SQLite database
└── keys/                           # Thư mục lưu khóa RSA
```

```

    └── {email}/
        ├── {email}_priv.enc      # Private key mã hóa AES
        └── {email}_pub.pem       # Public key PEM format
    └── logs/
        ├── security.log          # Log bảo mật
        └── signature_log.json     # Audit trail chữ ký số
    └── encrypted_files/         # Tệp tin mã hóa (tùy chọn)

```

## 4. THUẬT TOÁN BẢO MẬT SỬ DỤNG

### 4.1 Mã hóa mật khẩu

- **Thuật toán:** SHA-256 với random salt
- **Quy trình:**

```

salt = os.urandom(32) # 32 bytes random salt
passphrase_hash = hashlib.sha256(passphrase.encode() + salt).hexdigest()

```

### 4.2 Khóa RSA

- **Kích thước:** 2048 bit
- **Thời hạn:** 90 ngày tự động
- **Lưu trữ:** Private key mã hóa AES-256-GCM
- **Định dạng:** PEM standard

### 4.3 Mã hóa tệp tin (Hybrid Encryption)

- **Session Key:** AES-256-GCM (256 bit)
- **Key Transport:** RSA-2048 OAEP padding
- **Quy trình:**

```

# 1. Sinh AES session key
session_key = os.urandom(32)

# 2. Mã hóa file bằng AES-GCM
cipher = AES.new(session_key, AES.MODE_GCM)
ciphertext, tag = cipher.encrypt_and_digest(file_data)

# 3. Mã hóa session key bằng RSA
encrypted_session_key = rsa_public_key.encrypt(session_key, OAEP())

```

### 4.4 Chữ ký số

- **Thuật toán:** RSA-PSS với SHA-256
- **Quy trình:**

```
# 1. Tính hash file  
file_hash = hashlib.sha256(file_data).digest()  
  
# 2. Ký hash bằng RSA-PSS  
signature = rsa_private_key.sign(file_hash, PSS(), SHA256())
```

## 4.5 Multi-Factor Authentication (MFA)

- **Chuẩn:** TOTP (Time-based OTP) - RFC 6238
- **Thời gian:** 30 giây window
- **Thuật toán:** HMAC-SHA1
- **QR Code:** Tích hợp Google Authenticator hoặc Microsoft Authenticator

# 5. CHỨC NĂNG ĐÃ THỰC HIỆN

## 5.1 Đăng ký tài khoản

- **Giao diện:** Form đăng ký với validation
- **Dữ liệu:** Email, họ tên, ngày sinh, SĐT, địa chỉ, passphrase
- **Bảo mật:**
  - Kiểm tra email trùng lặp
  - Validation passphrase mạnh ( $\geq 8$  ký tự, hoa-thường-số-ký tự đặc biệt)
  - SHA-256 hash với random salt
- **Kết quả:** Lưu vào SQLite database với role 'user' mặc định

## 5.2 Đăng nhập & MFA

- **Xác thực:** Email + passphrase hash verification
- **MFA:** TOTP code từ Google Authenticator
- **Bảo mật:**
  - Giới hạn 5 lần đăng nhập sai
  - Khóa tài khoản tự động với lockout progressive
  - Session management an toàn
- **Logging:** Ghi log tất cả hoạt động authentication

## 5.3 Quản lý khóa RSA

- **Tạo khóa:** RSA-2048 bit với thời hạn 90 ngày
- **Lưu trữ:** Private key mã hóa AES-GCM, public key PEM
- **Quản lý:** Kiểm tra hết hạn, gia hạn, tạo mới
- **Xuất khóa:** Export PEM format và QR code

## 5.4 QR Code Public Key

- **Tạo QR:** Chứa email, ngày tạo, public key (base64)
- **Đọc QR:** Scan từ file ảnh hoặc camera
- **Chia sẻ:** Chia sẻ public key an toàn qua QR

## 5.5 Cập nhật tài khoản

- **Profile:** Sửa thông tin cá nhân (tên, ngày sinh, SĐT, địa chỉ)
- **Đổi passphrase:**
  - Xác thực passphrase cũ
  - Giải mã private key với passphrase cũ
  - Mã hóa lại private key với passphrase mới
- **MFA:** Bật/tắt TOTP authentication

## 5.6 Mã hóa tệp tin

- **Hybrid:** AES-256-GCM + RSA-2048
- **Metadata:** Thông tin người gửi, tên file, timestamp
- **Định dạng:**
  - Combined: File .enc chứa tất cả
  - Separate: File .enc + file .key riêng biệt

## 5.7 Giải mã tệp tin

- **Tự động:** Nhận diện định dạng file mã hóa
- **Xác thực:** Yêu cầu passphrase để giải mã private key
- **Khôi phục:** Giải mã thành công trả về file gốc
- **Kiểm tra:** Integrity verification với GCM tag

## 5.8 Ký số tệp tin

- **Thuật toán:** RSA-PSS + SHA-256
- **Output:** File .sig chứa chữ ký số
- **Audit:** Ghi log vào signature\_log.json
- **Metadata:** Thông tin người ký, thời gian, file hash

## 5.9 Xác minh chữ ký

- **Input:** File gốc + file .sig
- **Verification:** Kiểm tra với public key
- **Kết quả:**
  - Hợp lệ: Hiển thị thông tin người ký
  - Không hợp lệ: Cảnh báo bị thay đổi
- **Logging:** Ghi log tất cả hoạt động verification

## 5.10 Phân quyền tài khoản

- **Roles:** 'user' và 'admin'
- **Admin functions:**
  - Xem danh sách users
  - Promote/demote users
  - Lock/unlock accounts
  - Xem system logs
- **UI:** Admin dashboard riêng biệt

5.11 Ghi log bảo mật

- **File:** `data/logs/security.log`
- **Format:** Timestamp, Email, Action, Status
- **Events:** Login, key generation, encryption, signing, admin actions
- **Audit:** `signature_log.json` cho digital signatures

 5.13 Kiểm tra trạng thái khóa

- **Thông tin:** Ngày tạo, hết hạn, trạng thái
- **Cảnh báo:** Thông báo gần hết hạn
- **Action:** Gia hạn hoặc tạo khóa mới
- **UI:** Giao diện trực quan cho key management

 5.14 Tìm kiếm public key

- **Search:** Tìm theo email address
- **Display:** Hiển thị public key, QR code, thông tin khóa
- **Import:** Nhập public key của users khác
- **Status:** Kiểm tra tính hợp lệ và hết hạn

 5.15 Giới hạn đăng nhập

- **Limit:** 5 lần đăng nhập sai
- **Lockout:** Progressive lockout (15min, 30min, 1hr, 2hr, 4hr)
- **Tracking:** Đếm fail\_count trong database
- **Auto-unlock:** Tự động mở khóa khi hết thời gian

 5.16 Tùy chọn định dạng file

- **Combined:** Tất cả trong 1 file .enc
- **Separate:** File .enc + file .key riêng
- **Auto-detect:** Tự động nhận diện khi giải mã
- **Metadata:** Thông tin format trong file header

 5.17 Khôi phục tài khoản

- **Recovery Code:** Tạo khi đăng ký (chỉ hiển thị 1 lần)
- **Process:** Nhập recovery code → đổi passphrase mới
- **Security:** Recovery code hash trong database
- **Re-encryption:** Giải mã và mã hóa lại private key

---

## 6. GIAO DIỆN NGƯỜI DÙNG

### 6.1 Màn hình chính



- **Tabs:** Login, Register, Keys, Encrypt, Decrypt, Sign, Verify, Account, Admin
- **Navigation:** Tab-based interface với back callbacks
- **Responsive:** Tự động điều chỉnh kích thước window

## 6.2 Đăng ký tài khoản

Computer Security Project 1

Email: tester2@gmail.com

Full Name: tester

Date of Birth: 2025-07-15

Phone: 0987654321

Address: 123 testing

Passphrase: \*\*\*\*\*

Confirm Passphrase: \*\*\*\*\*

Show Passphrase

**Continue**

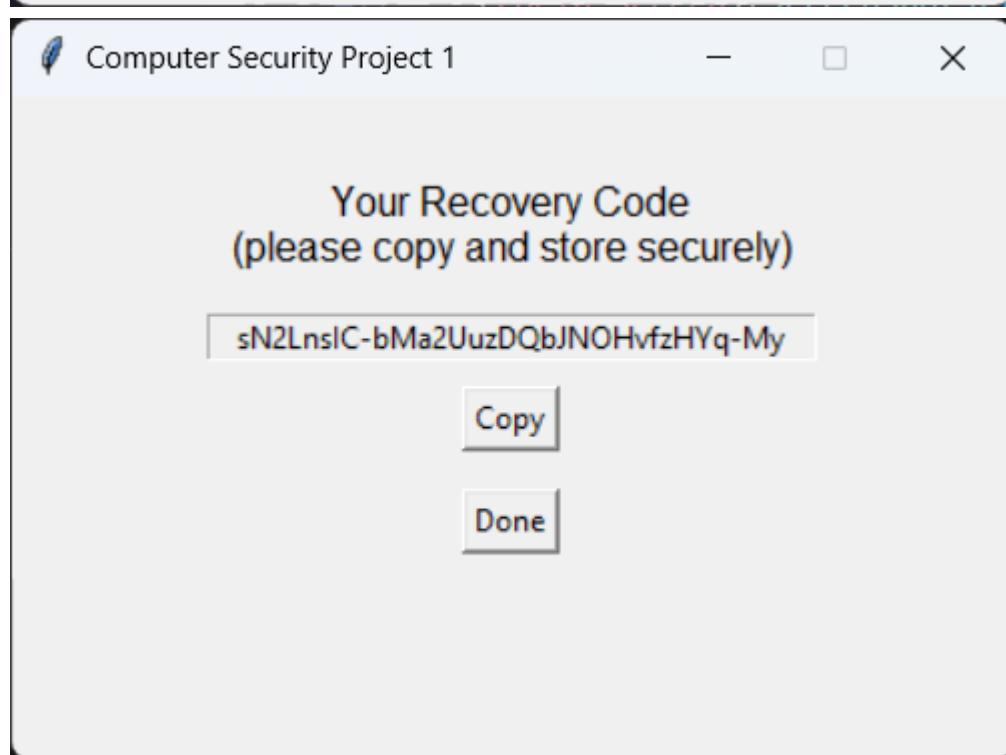
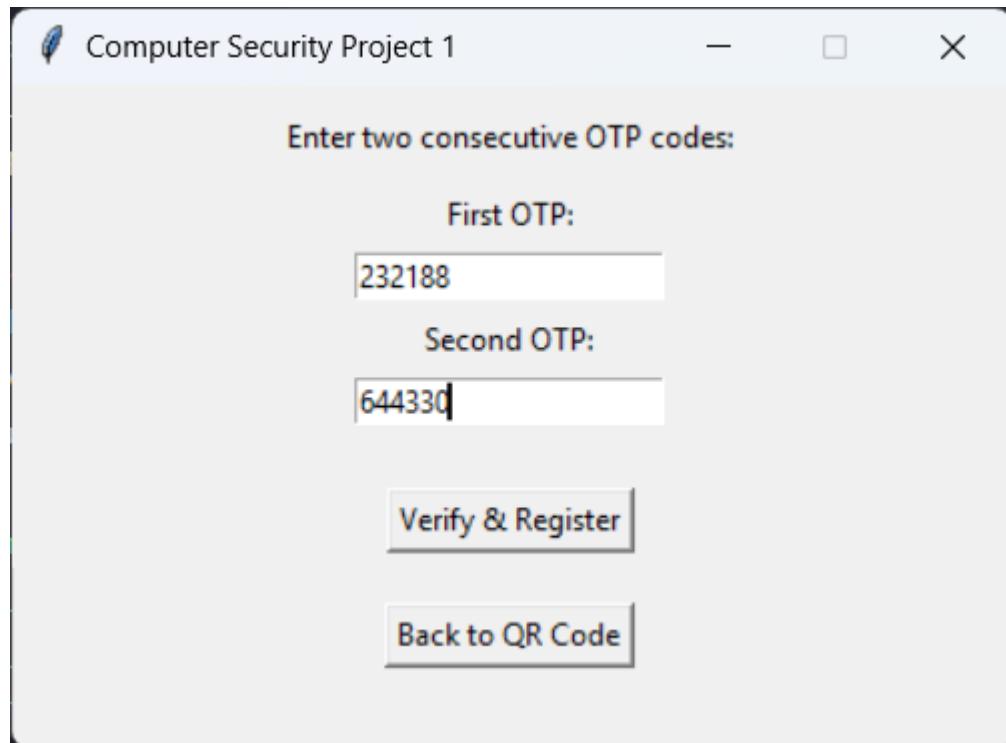
**Back**





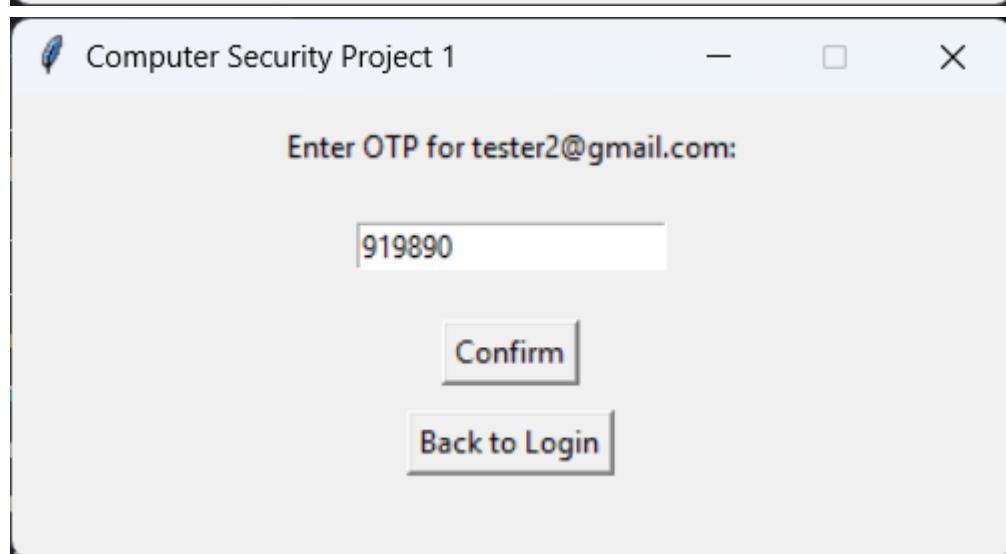
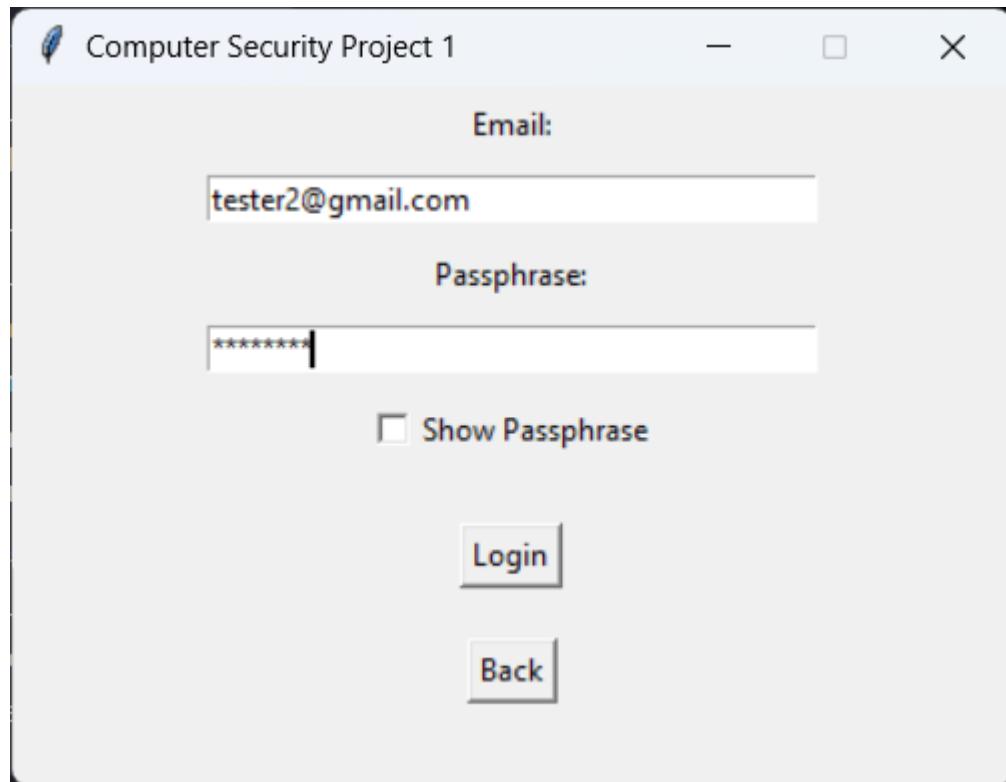
[Next: Verify OTP](#)

[Back](#)



- **Fields:** Email, full name, date of birth, phone, address, passphrase
- **Validation:** Kiểm tra email hợp lệ, passphrase mạnh
- **MFA:** Tạo TOTP secret key
- **Recovery Code:** Hiển thị 1 lần duy nhất sau khi đăng ký thành công

## 6.3 Đăng nhập & MFA



- **Fields:** Email, passphrase
- **MFA:** TOTP code input
- **Security:** Hiển thị lockout status

#### 6.4 Thay đổi thông tin tài khoản

Computer Security Project 1

## Update Account (cakey@gmail.com)

Full Name:

Date of Birth:

Phone:

Address:

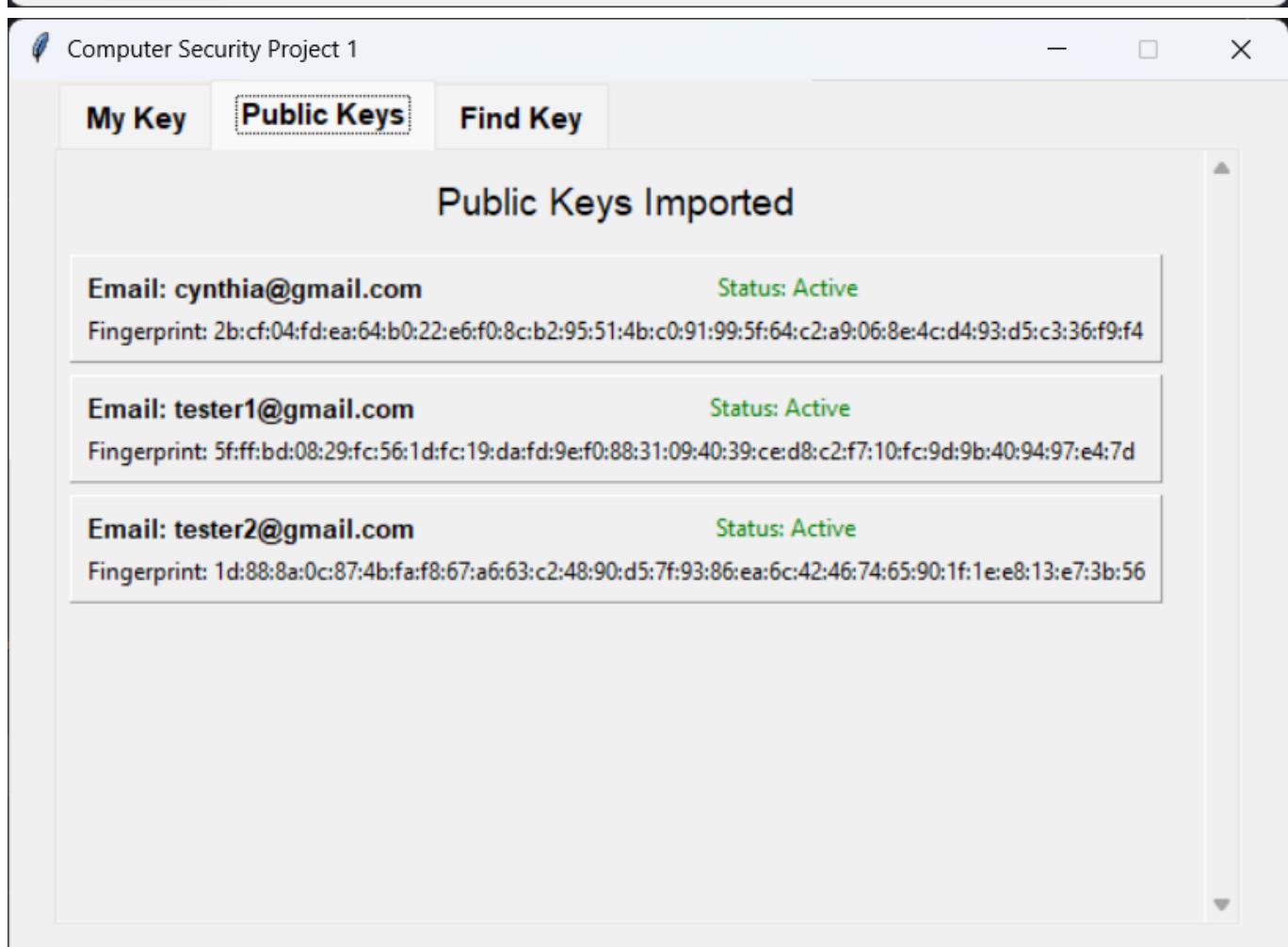
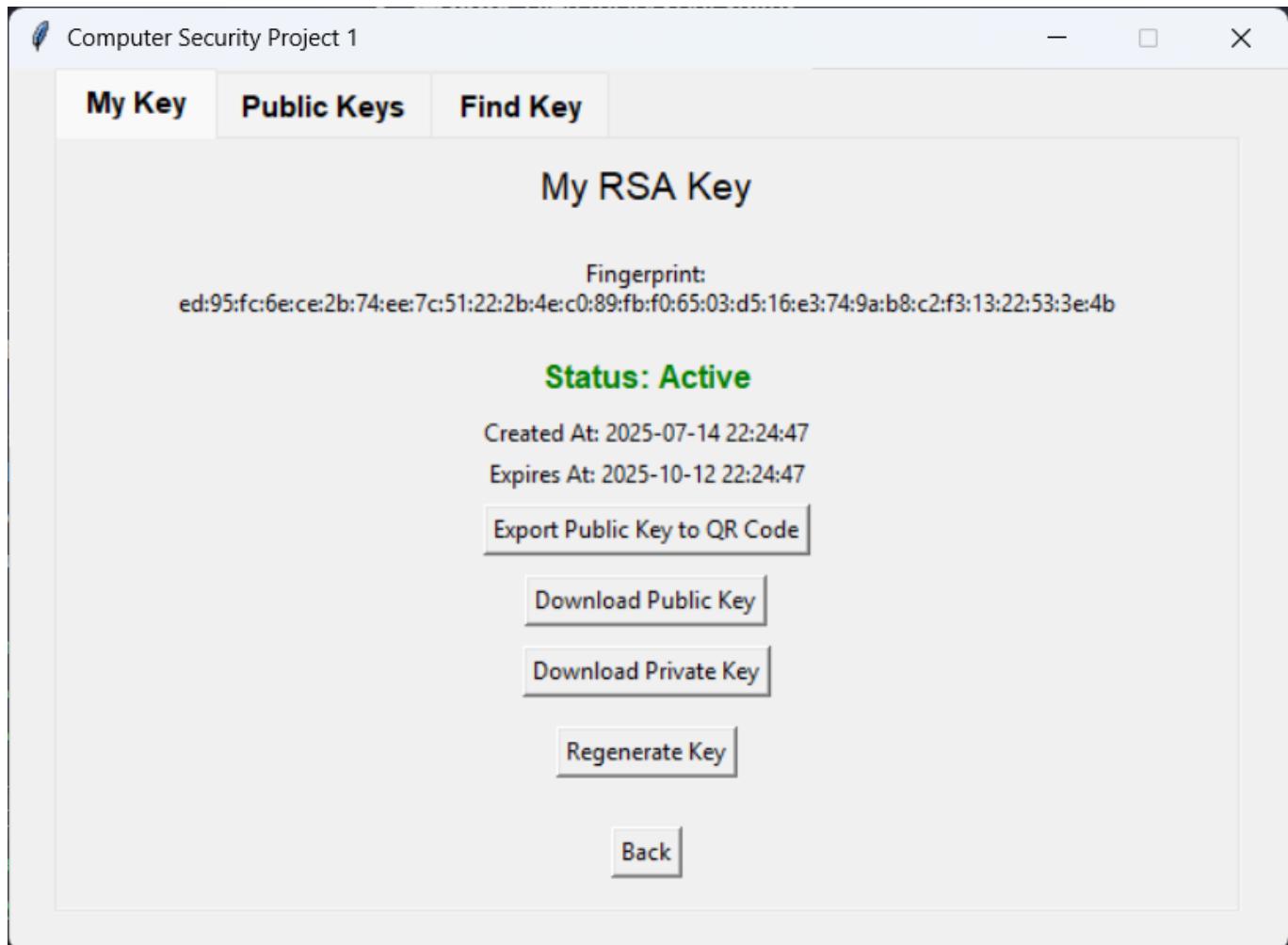
**Save Changes**

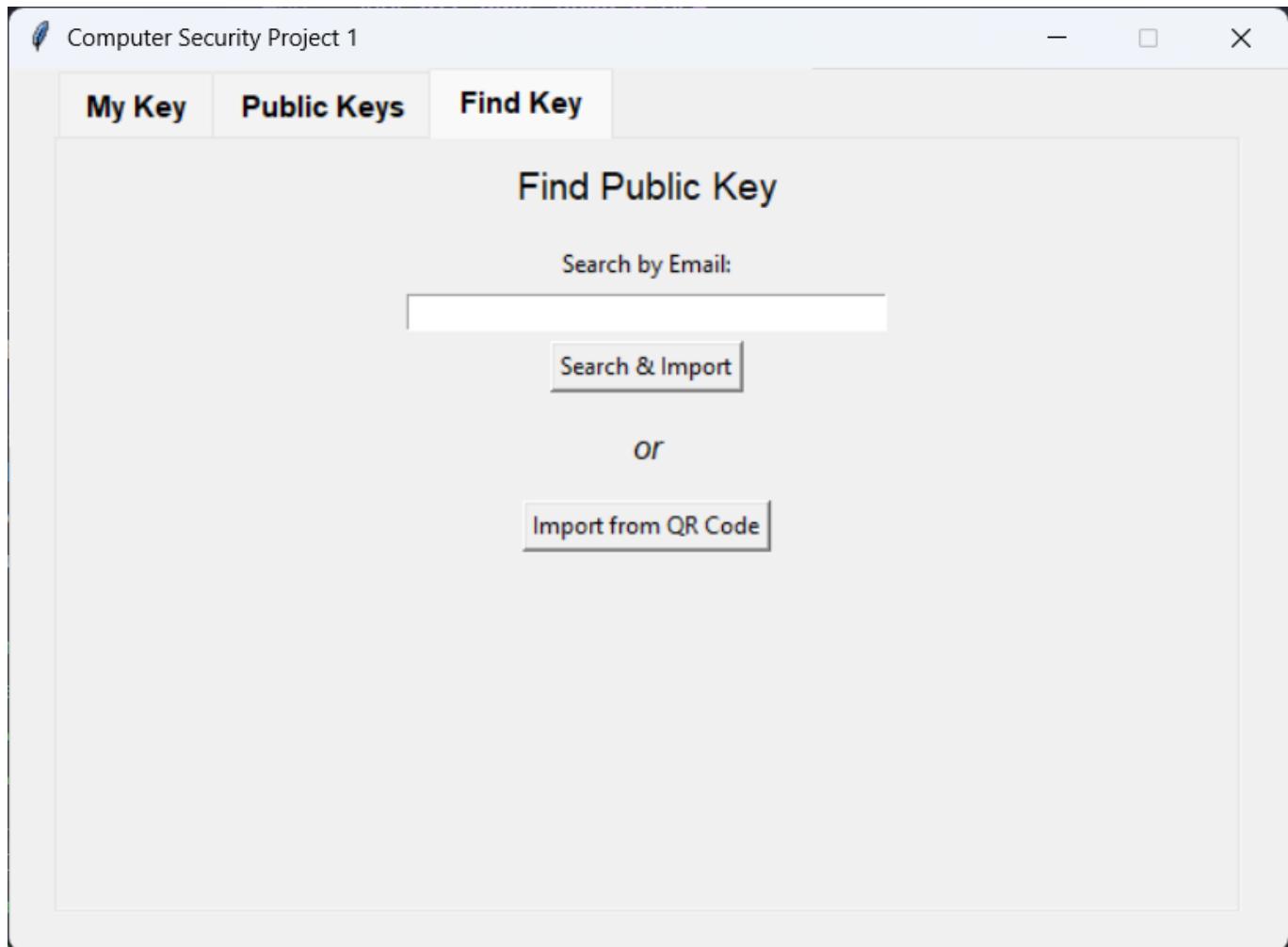
**Change Passphrase**

**Back**

- **Fields:** Full name, date of birth, phone, address
- **Passphrase:** Thay đổi passphrase

## 6.5 Quản lý khóa

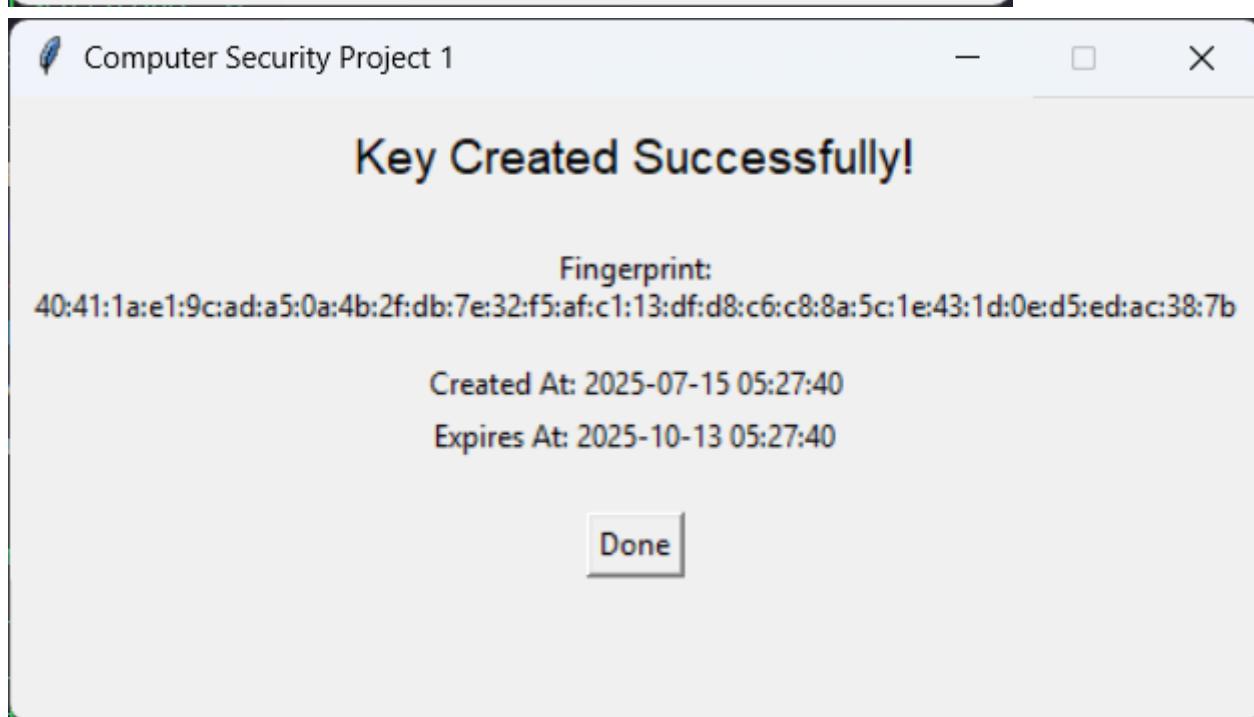
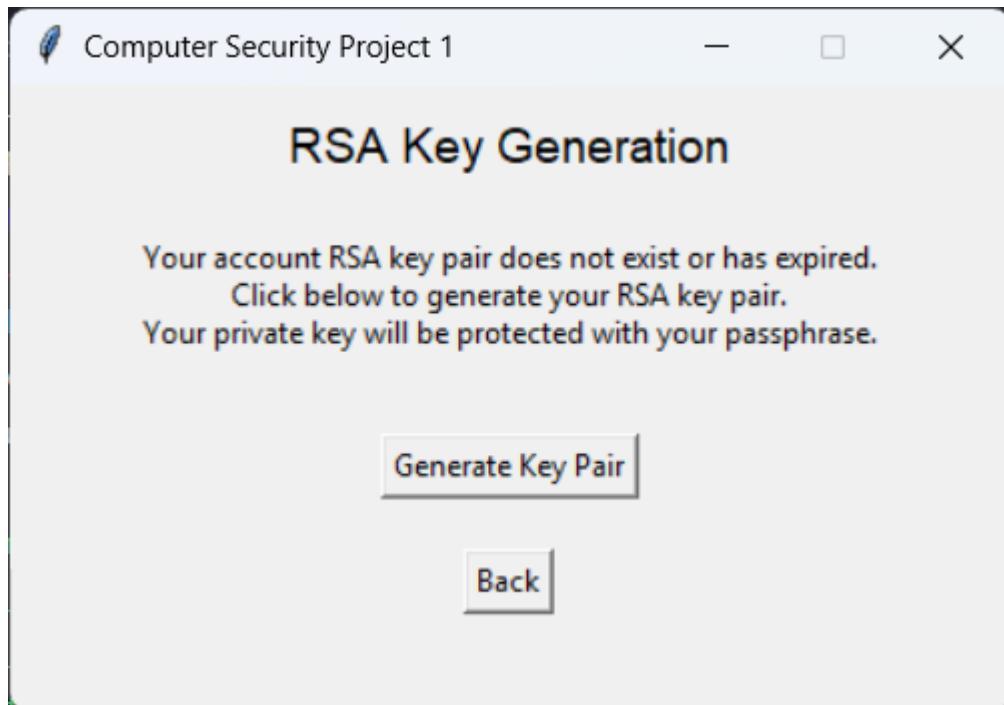




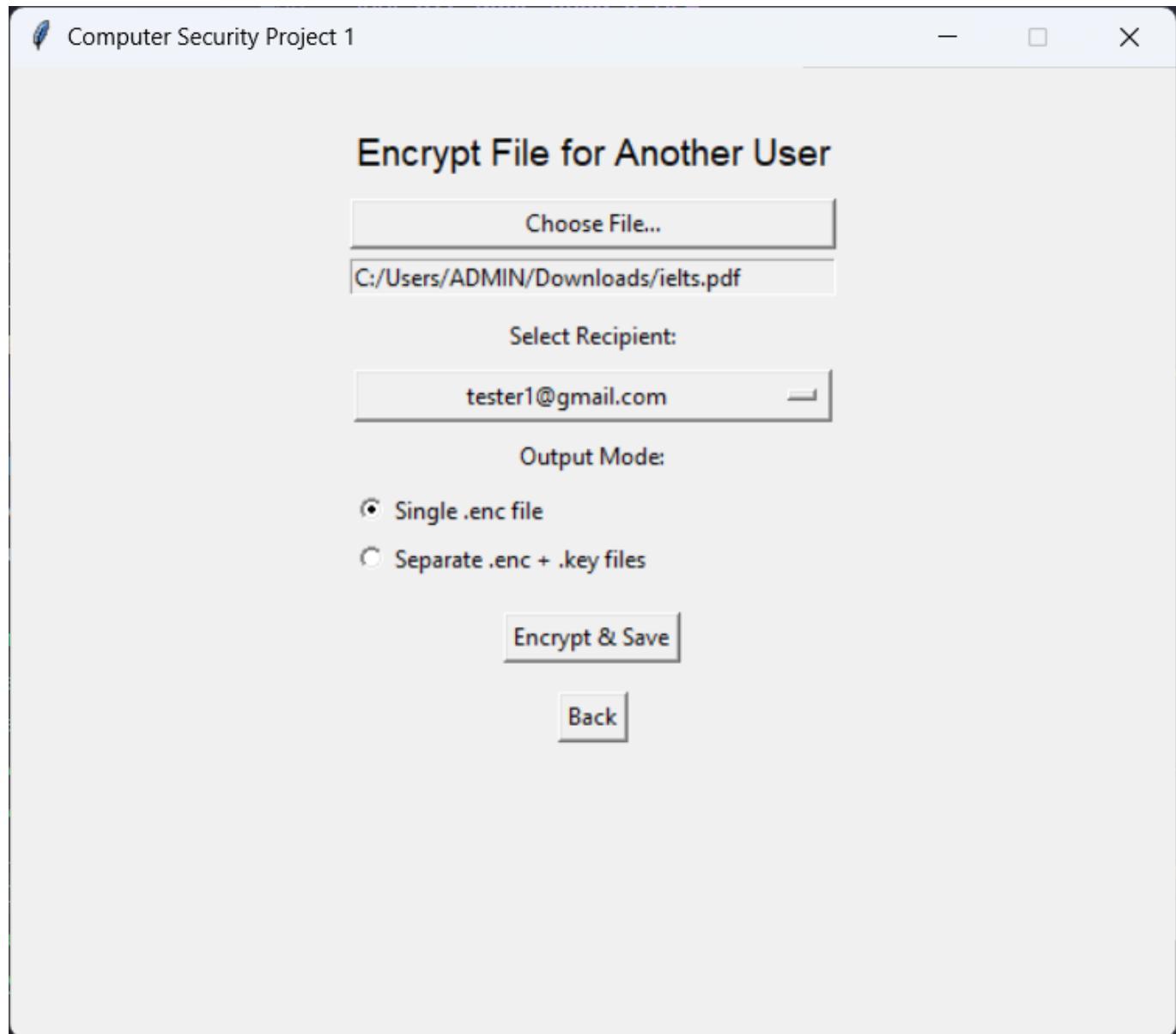


- **Status:** Thông tin khóa hiện tại
- **Actions:** Tạo mới, tải về
- **QR Code:** Hiển thị QR code cho public key
- **Public Keys:** Xem danh sách public keys đã lưu
- **Find Key:** Tìm kiếm public key theo email hoặc nạp từ QR

## 6.6 Tạo khóa RSA

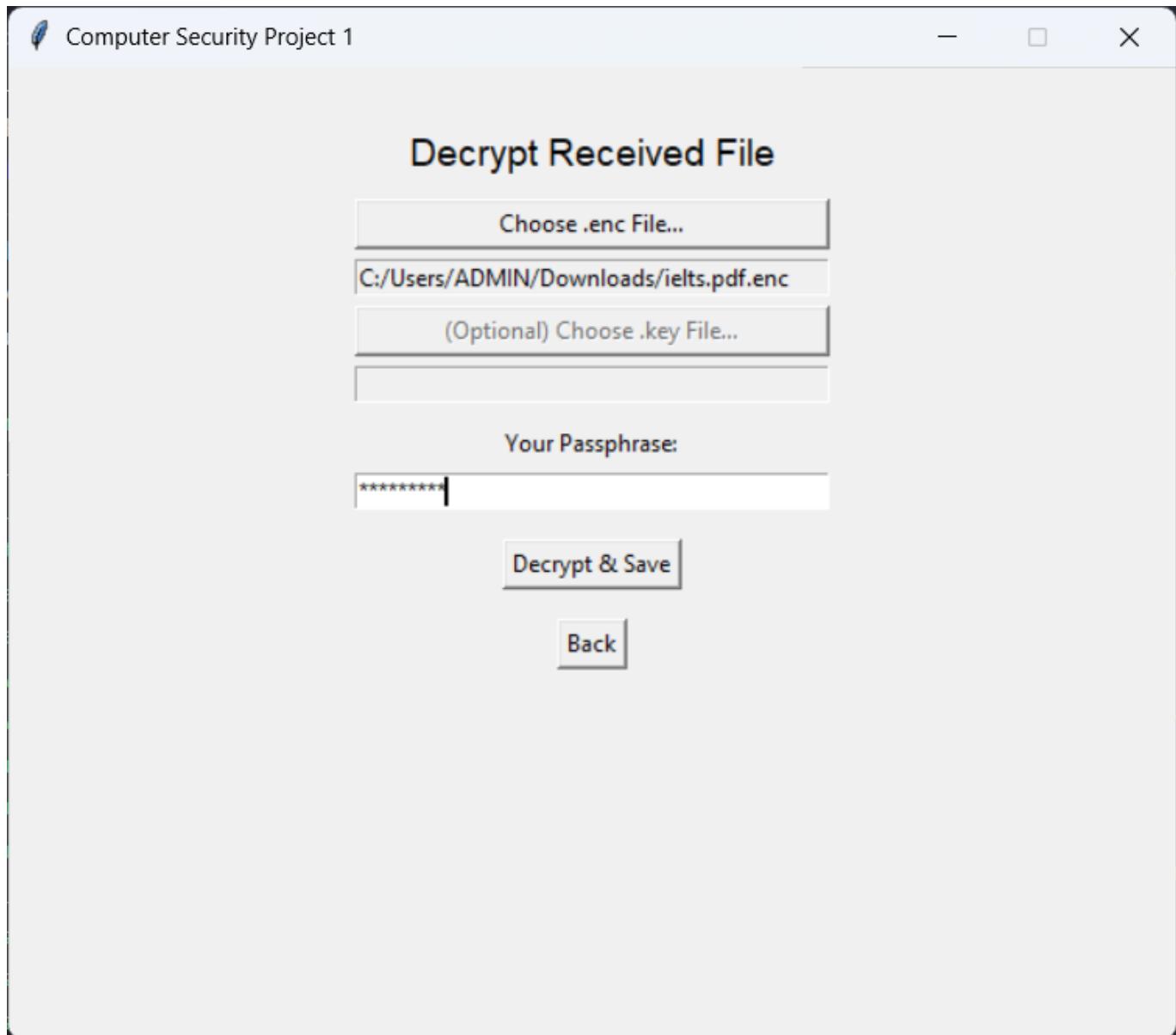


## 6.7 Mã hóa tệp tin



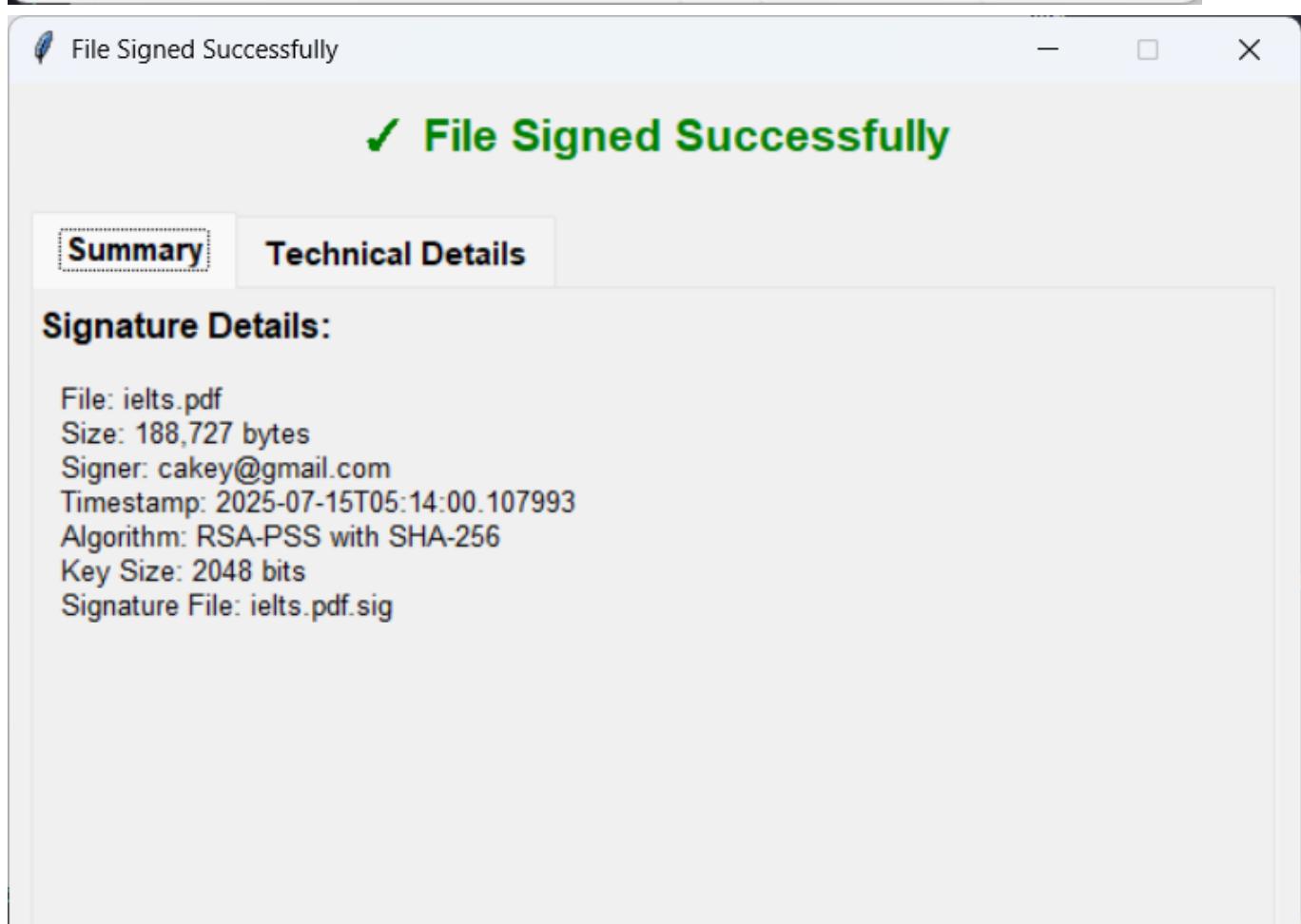
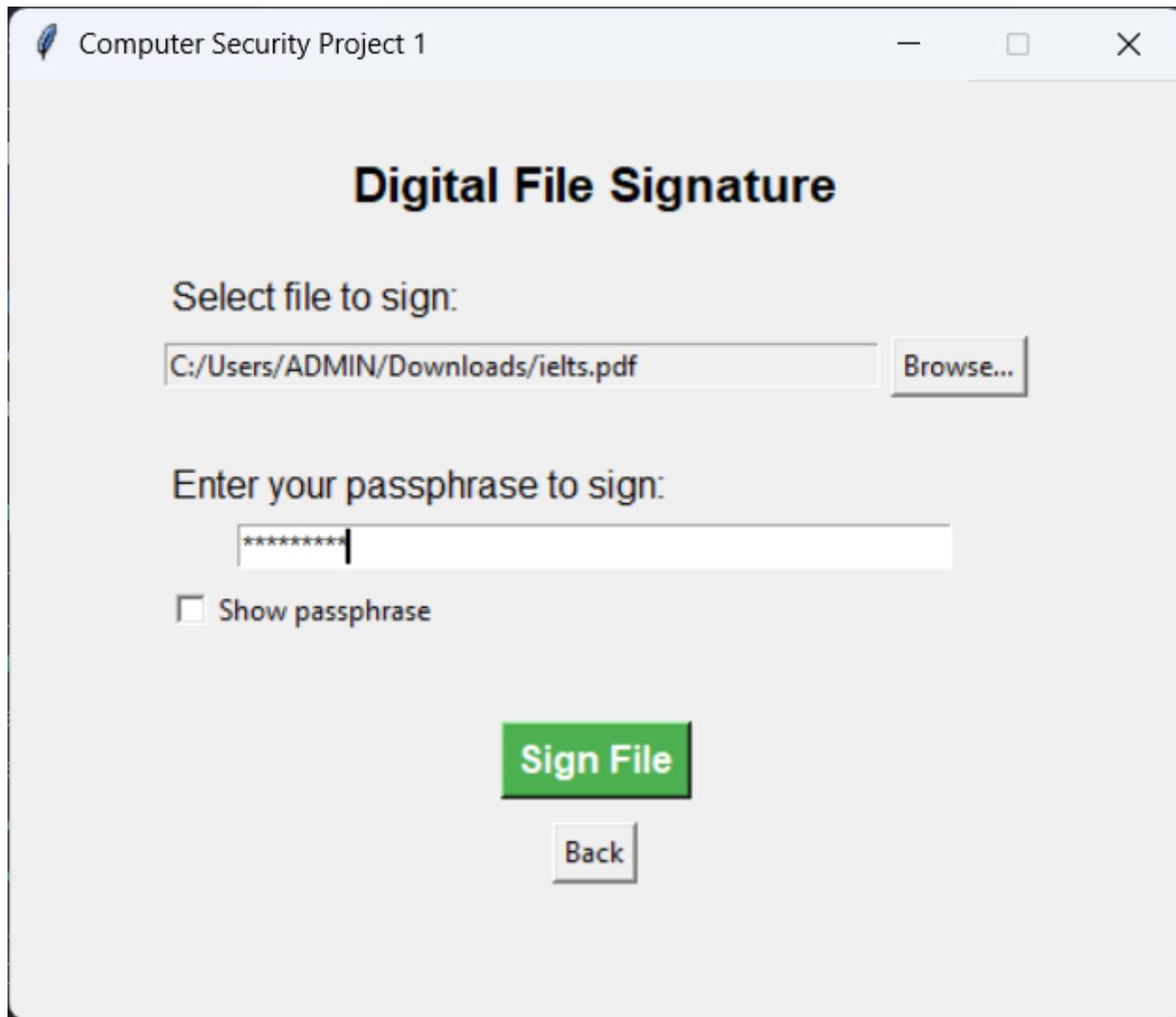
- **File Selection:** Browse file để mã hóa
- **Recipient:** Chọn public key người nhận
- **Options:** Combined/separate format

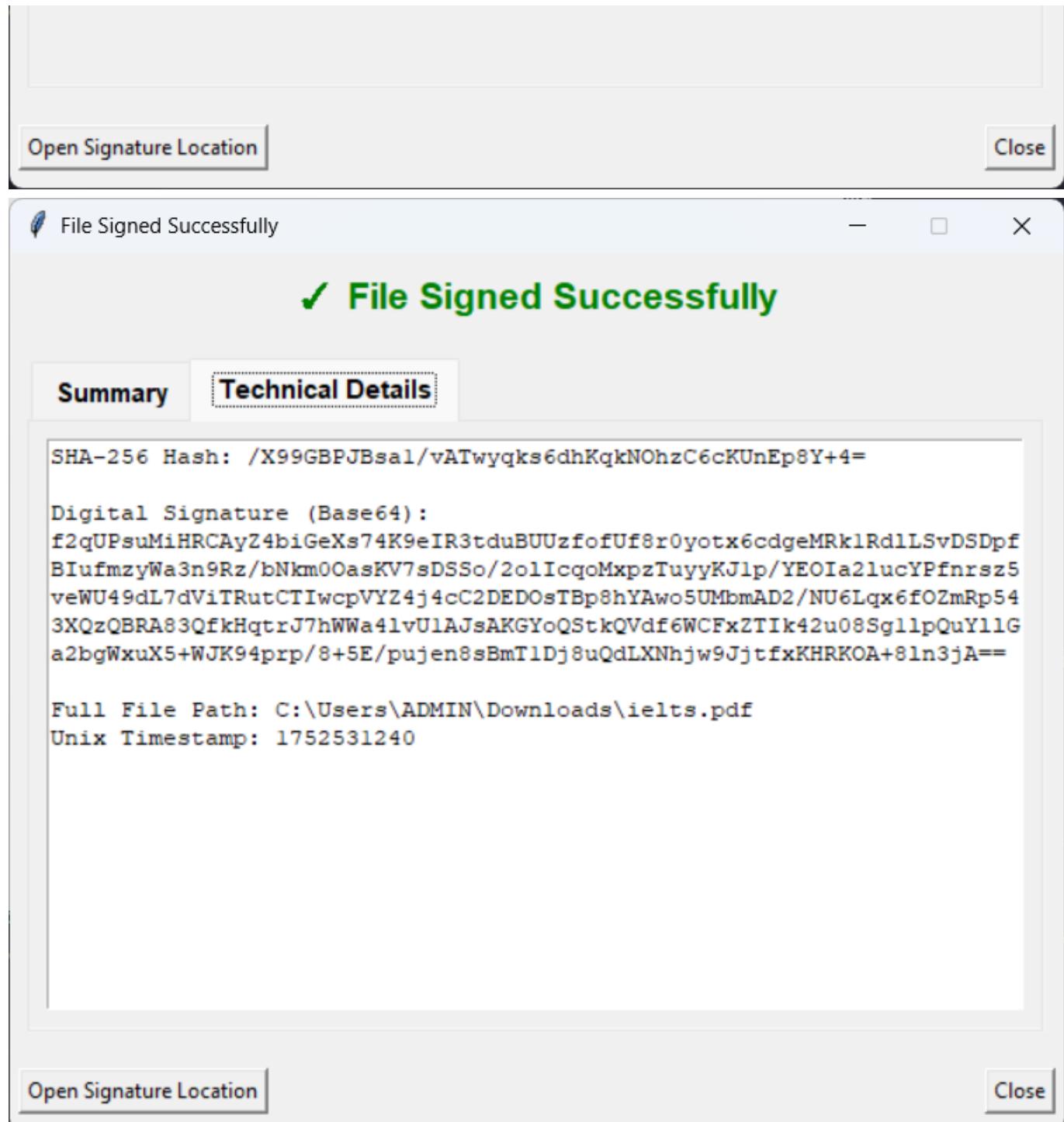
## 6.8 Giải mã tệp tin



- **File Selection:** Browse file để giải mã
- **Key File:** Chọn file chứa session key (nếu separate)
- **Passphrase:** Nhập passphrase để giải mã private key
- **Output:** Lưu file giải mã vào thư mục chọn

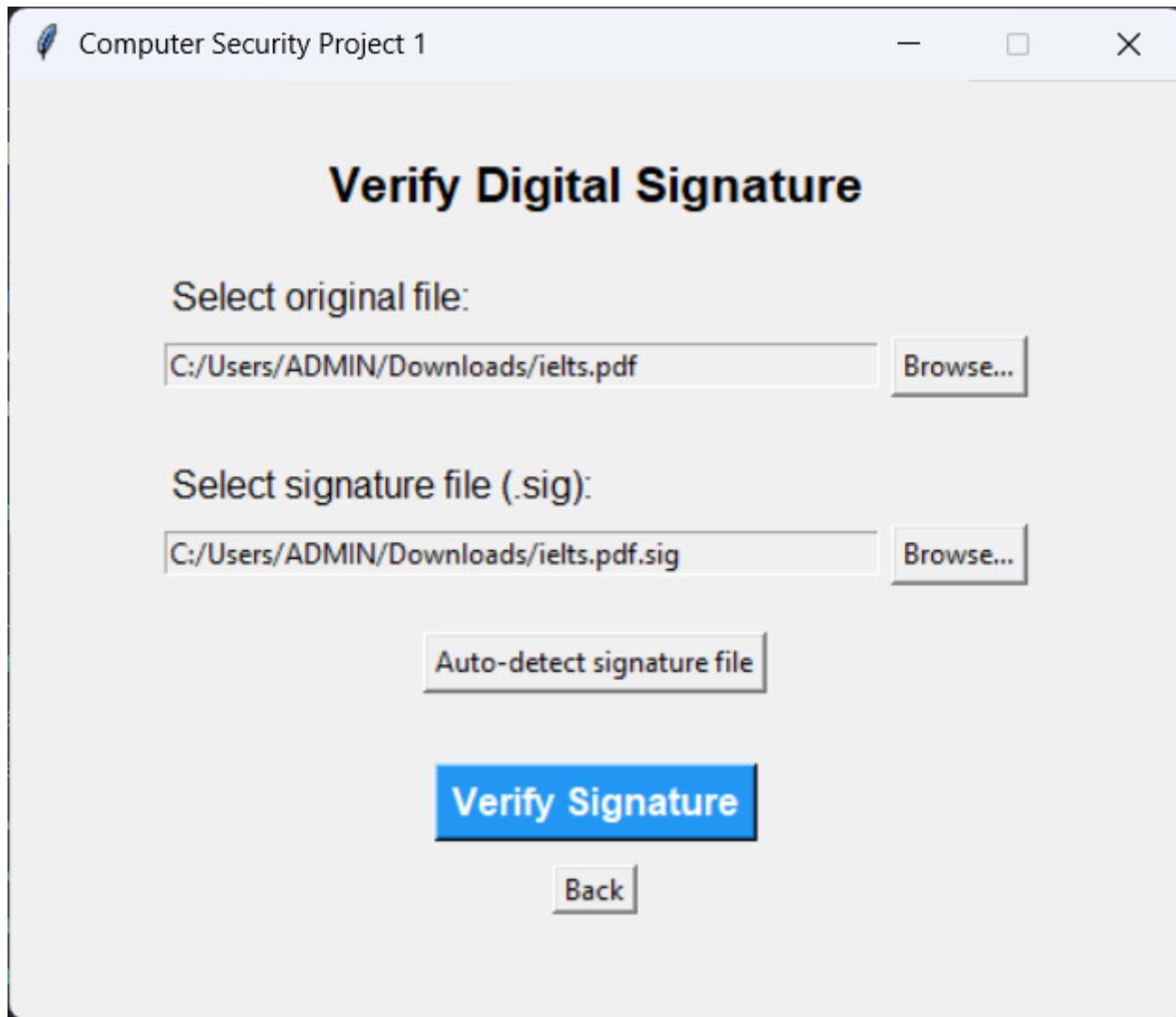
## 6.9 Ký số tệp tin



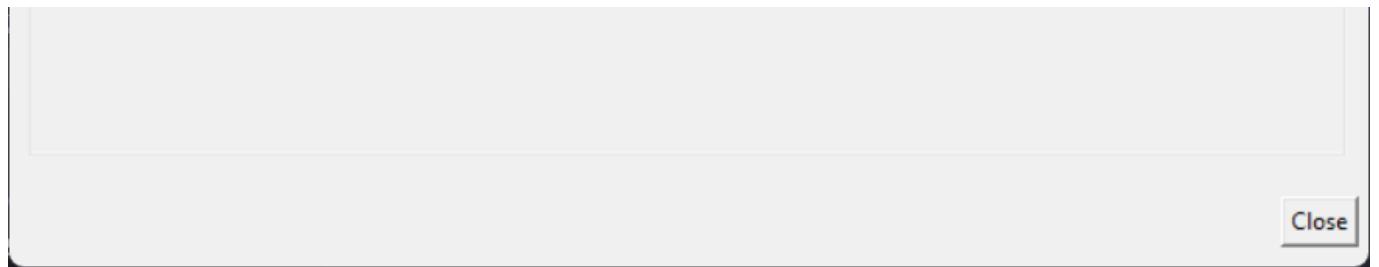


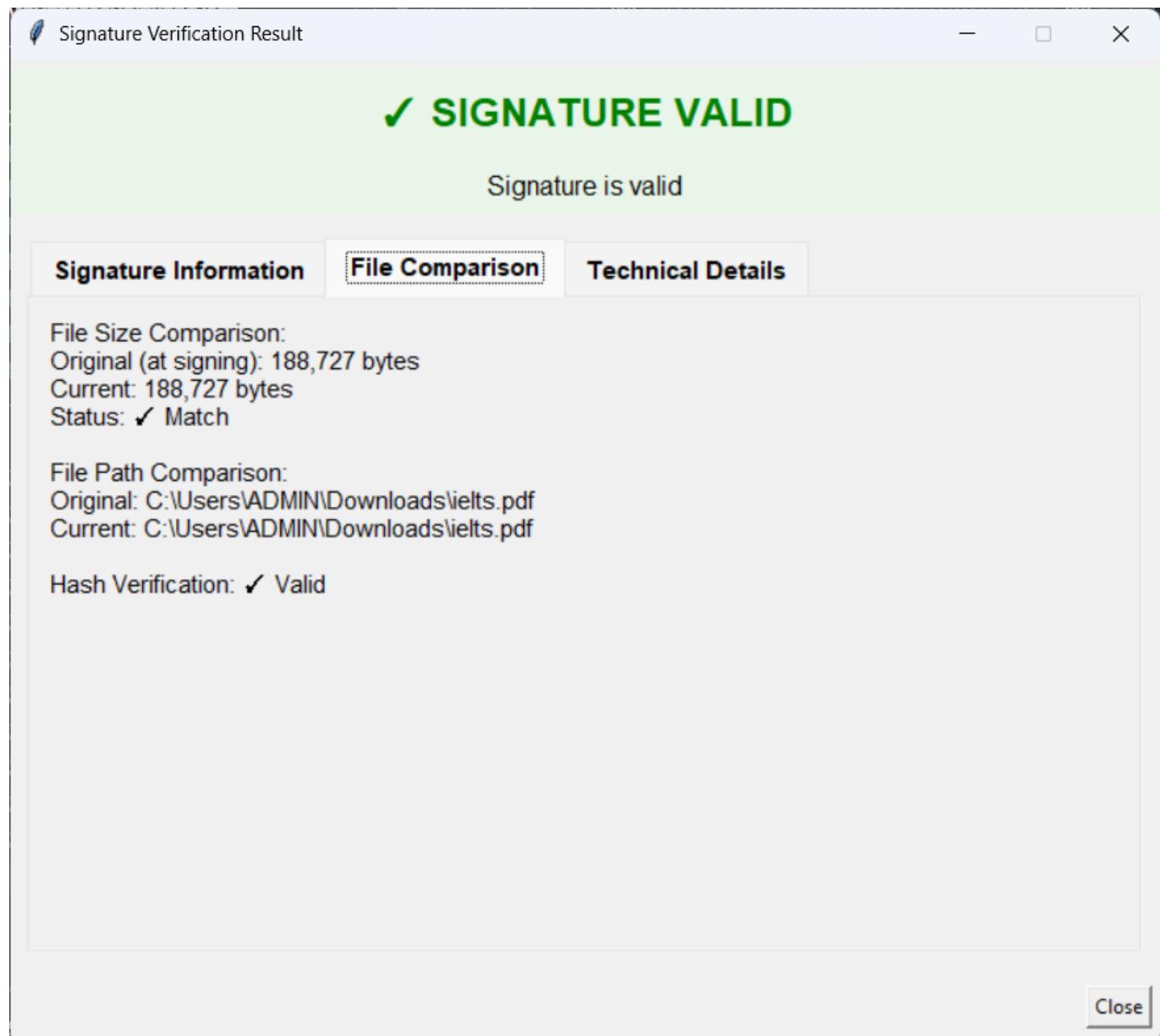
- **File Selection:** Browse file để ký
- **Signature Output:** Lưu chữ ký vào file .sig và xem thông tin
- **Passphrase:** Nhập passphrase để ký

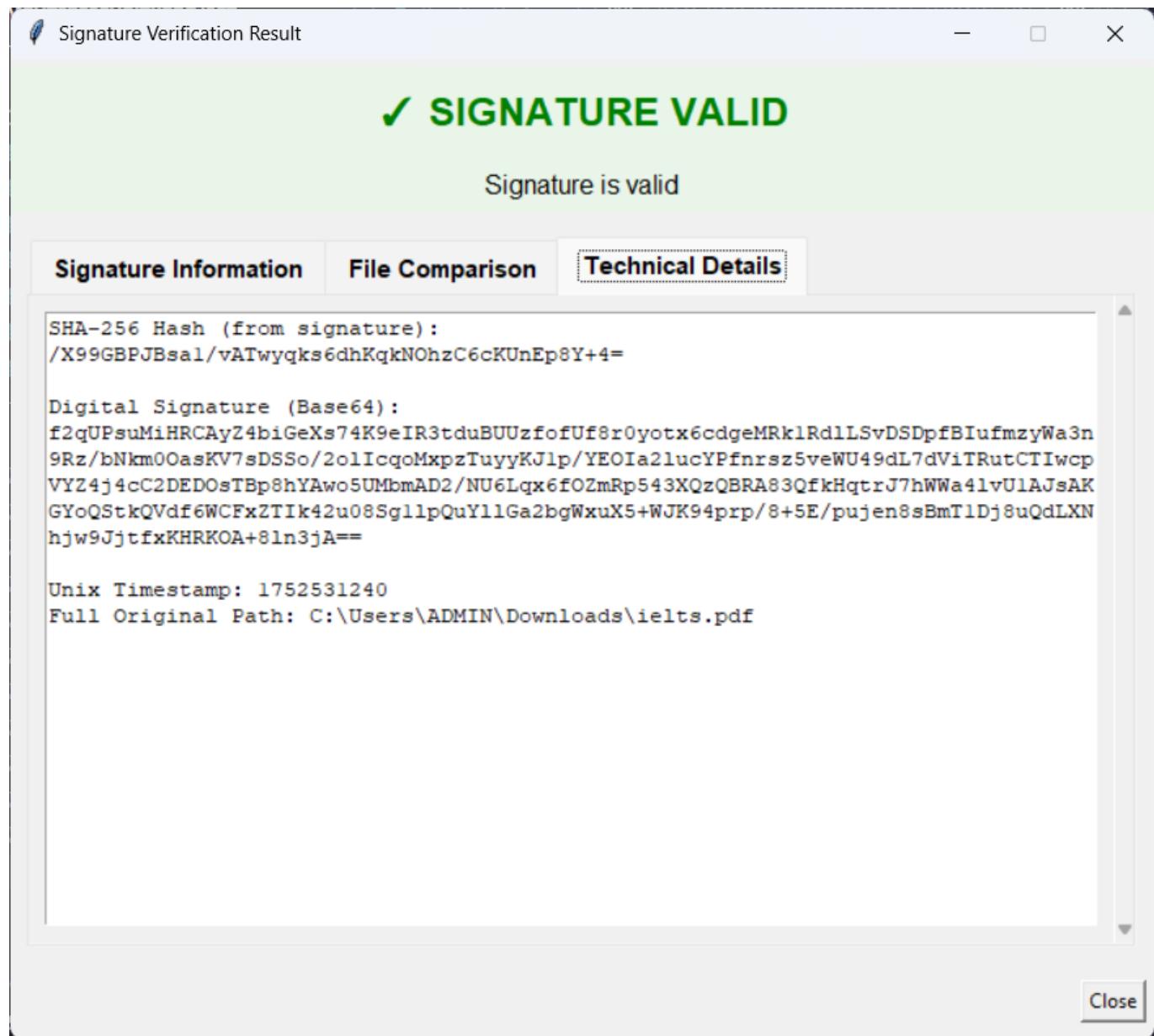
## 6.10 Xác minh chữ ký



The screenshot shows a window titled "Signature Verification Result". It displays a green banner with "✓ SIGNATURE VALID" and the message "Signature is valid". Below this are three tabs: "Signature Information" (selected), "File Comparison", and "Technical Details". Under "Signature Information", the following details are listed:  
Signed File: ielts.pdf  
Original File Size: 188,727 bytes  
Signer: cakey@gmail.com  
Signed On: 2025-07-15T05:14:00.107993  
Algorithm: RSA-PSS with SHA-256  
Key Size: 2048 bits  
Verification Message: Signature is valid







- **File Selection:** Browse file gốc và file chữ ký
- **Verification Result:** Hiển thị thông tin người ký, thời gian, trạng thái, chi tiết

## 6.11 Admin Dashboard

Computer Security Project 1

## Admin Dashboard

**Users** **Logs**

Email	Role	Locked
cakey@gmail.com	admin	No
cynthia@gmail.com	admin	No
tester1@gmail.com	user	Yes

[Lock Account](#)  
[Unlock Account](#)  
[Promote to Admin](#)  
[Demote to User](#)

[Back](#)

Computer Security Project 1

## Admin Dashboard

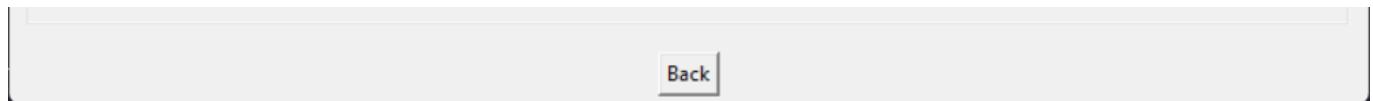
**Users** **Logs**

### System Log

```

2025-07-14 22:24:36,003 | INFO | User 'cakey@gmail.com' registered successfully.
2025-07-14 22:24:37,321 | INFO | Recovery code for 'cakey@gmail.com' saved successfully.
2025-07-14 22:24:47,563 | INFO | User 'cakey@gmail.com' generated RSA key pair
2025-07-14 22:25:08,077 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 22:35:05,743 | INFO | User 'cynthia@gmail.com' registered successfully.
2025-07-14 22:35:07,581 | INFO | Recovery code for 'cynthia@gmail.com' saved successfully.
2025-07-14 22:36:18,761 | INFO | User 'cynthia@gmail.com' logged in successfully.
2025-07-14 22:36:23,056 | INFO | User 'cynthia@gmail.com' generated RSA key pair
2025-07-14 22:40:06,183 | INFO | User 'tester1@gmail.com' registered successfully.
2025-07-14 22:40:08,018 | INFO | Recovery code for 'tester1@gmail.com' saved successfully.
2025-07-14 22:40:26,628 | INFO | User 'tester1@gmail.com' generated RSA key pair
2025-07-14 22:40:46,694 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 22:48:31,355 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 22:52:24,437 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 22:52:35,501 | INFO | Admin cakey@gmail.com promoted 'cynthia@gmail.com'
2025-07-14 23:02:10,092 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 23:03:13,212 | INFO | User 'cakey@gmail.com' logged in successfully.
2025-07-14 23:04:16,481 | INFO | Admin cakey@gmail.com locked 'tester1@gmail.com'
2025-07-15 04:58:37,589 | INFO | User 'cakey@gmail.com' logged in successfully.

```

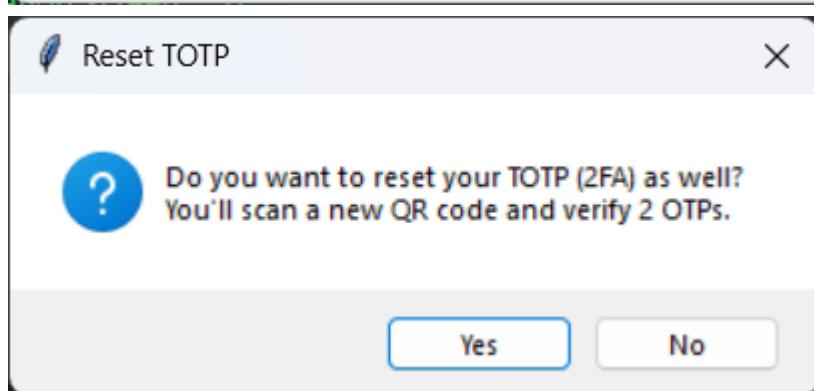


- **User List:** Danh sách users với role và status
- **Actions:** Promote, demote, lock, unlock
- **Logs:** System security logs viewer

## 6.12 Account Recovery

The screenshot shows a window titled "Computer Security Project 1" with a tab labeled "Recover Account". The form contains the following fields:

- Email: tester2@gmail.com
- Recovery Code: sN2LnsIC-bMa2UuzDQbJNOHvfzHYq-My
- New Passphrase: \*\*\*\*\*
- Confirm New Passphrase: \*\*\*\*\*
- Show Passphrase:
- Recover: A large blue button.
- Back: A smaller button below the Recover button.



- **Recovery Code Input:** Nhập recovery code để khôi phục tài khoản
- **New Passphrase:** Nhập passphrase mới

- **MFA Reset:** Tùy chọn đặt lại TOTP nếu cần
- 

## 7. TESTING & VALIDATION

### 7.1 Unit Tests

```
# Test RSA key generation
def test_rsa_key_generation():
    private_key, public_key = generate_rsa_key_pair()
    assert private_key is not None
    assert public_key is not None
    assert len(private_key.exportKey()) > 0

# Test file encryption/decryption
def test_file_encryption():
    # Tạo file test
    test_data = b"Hello, World!"
    encrypted = encrypt_file_for_user("test@example.com", public_key, test_data)
    decrypted = decrypt_file_for_user("test@example.com", private_key, encrypted)
    assert decrypted == test_data

# Test digital signature
def test_digital_signature():
    signature = sign_file_data(test_data, private_key)
    is_valid = verify_file_signature(test_data, signature, public_key)
    assert is_valid == True
```

### 7.2 Security Tests

- **Passphrase Strength:** Validation các yêu cầu mật khẩu mạnh
- **Encryption Strength:** Verify AES-256-GCM và RSA-2048
- **Key Expiration:** Test tự động hết hạn khóa
- **Account Lockout:** Test progressive lockout mechanism
- **SQL Injection:** Parameterized queries protection

### 7.3 Performance Tests

- **Key Generation:** ~1-2 giây cho RSA-2048
  - **File Encryption:** Linear scaling với file size
  - **Database:** Optimized queries với proper indexing
  - **Memory Usage:** Efficient memory management
- 

## 8. BẢO MẬT & BEST PRACTICES

### 8.1 Cryptographic Standards

- **AES-256-GCM:** NIST approved, authenticated encryption
- **RSA-2048:** Current industry standard

- **SHA-256:** Secure hashing algorithm
- **PBKDF2:** 100,000 iterations for key derivation
- **TOTP:** RFC 6238 compliant

## 8.2 Security Measures

- **Input Validation:** Sanitize user inputs
- **SQL Injection Protection:** Parameterized queries
- **Memory Protection:** Clear sensitive data after use
- **File Permissions:** Restricted access to key files
- **Session Management:** Secure session handling

## 8.3 Audit & Logging

- **Comprehensive Logging:** Tất cả security events
- **Audit Trail:** Tamper-evident signature logs
- **Access Control:** Role-based permissions
- **Monitoring:** Real-time security monitoring

---

# 9. DEPLOYMENT & USAGE

## 9.1 System Requirements

- **OS:** Windows 10+, macOS 10.15+, Linux (Ubuntu 18.04+)
- **Python:** 3.12+ (recommended 3.12.10)
- **Memory:** 4GB RAM minimum
- **Storage:** 100MB free space

## 9.2 Installation

```
# Clone repository
git clone https://github.com/Burncake/ComputerSecurityProject.git
cd ComputerSecurityProject

# Setup virtual environment
python -m venv venv
venv\Scripts\Activate.ps1

# Install dependencies
pip install -r requirements.txt

# Run application
python main.py
```

## 9.3 First Time Setup

1. **Database Init:** Tự động tạo SQLite database
2. **Admin Account:** Đăng ký tài khoản đầu tiên và chỉnh role thành admin trực tiếp trong database

3. **Key Generation:** Tạo RSA key pair
  4. **MFA Setup:** Cấu hình TOTP authentication
- 

## 10. DEMO

Link to Demo Video on [Google Drive](#)

Link to Demo Video on [OneDrive](#)

---

## 11. KẾT LUẬN

### 11.1 Thành tựu đạt được

- **Hoàn thành ~94%** các yêu cầu bắt buộc (16/17 chức năng)
- **Bảo mật cao** với industry-standard algorithms
- **Giao diện thân thiện** với Tkinter GUI
- **Kiến trúc rõ ràng** và dễ maintain
- **Documentation đầy đủ** và chi tiết

### 11.2 Điểm nổi bật

- **Hybrid Encryption:** Kết hợp AES + RSA hiệu quả
- **MFA Implementation:** TOTP chuẩn RFC 6238
- **Key Lifecycle:** Quản lý khóa tự động với expiration
- **Audit Trail:** Comprehensive security logging
- **Role-based Access:** Phân quyền admin/user rõ ràng

### 11.3 Hướng phát triển tương lai

- **Cloud Integration:** Đồng bộ keys across devices
- **Mobile App:** Companion app cho MFA
- **Hardware Security:** HSM integration
- **PKI Infrastructure:** Certificate authority
- **Advanced Crypto:** Post-quantum algorithms

### 11.4 Bài học kinh nghiệm

- **Security First:** Luôn ưu tiên bảo mật trong design
  - **User Experience:** Cân bằng giữa security và usability
  - **Testing:** Comprehensive testing cho crypto operations
  - **Documentation:** Importance of clear documentation
- 

## 12. TÀI LIỆU THAM KHẢO

### 12.1 Cryptographic Standards

- [NIST SP 800-132](#) - PBKDF2 Recommendations

- [RFC 8017 - PKCS #1 v2.2: RSA Cryptography](#)
- [RFC 6238 - TOTP Algorithm](#)
- [FIPS 197 - AES Specification](#)

## 12.2 Security Best Practices

- [OWASP Cryptographic Storage Cheat Sheet](#)
- [Python Cryptography Documentation](#)
- [NIST Cybersecurity Framework](#)

## 12.3 Libraries & Tools

- [pycryptodome Documentation](#)
  - [PyOTP Documentation](#)
  - [Tkinter Documentation](#)
  - [SQLite Documentation](#)
- 

 **Ghi chú:** Báo cáo này được tạo ngày 15/7/2025 cho đồ án An ninh máy tính 1. Toàn bộ source code và documentation có sẵn tại [GitHub Repository](#).

---

 **Cảnh báo Bảo mật:** Ứng dụng này xử lý dữ liệu nhạy cảm và keys mã hóa. Luôn chạy trên hệ thống tin cậy và cập nhật thường xuyên. Đối với sử dụng production, cần thực hiện đánh giá bảo mật và penetration testing đầy đủ.