Everest Membership System 보안 구현 증명서

1. 보안 아키텍처 개요 (Security Architecture)

본 시스템은 **2**중 보안 계층을 적용하여 관리자 권한과 회원 개인정보를 강력하게 보호하고 있습니다.

보호 대상	적용 기술	설명
관리자 비밀번호	**SHA-256 Hashing**	단방향 암호화 기술로, 해커가 DB를 탈취해도 원래 비밀번호를 복원할 수 없습니다. (256비트 해시)
회원 개인정보	**Fernet Encryption**	**AES-128 (CBC)** 암호화와 **HMAC-SHA256** 무결성 인증을 결합한 산업 표준 암호화 방식입니다. (256비트 키 사용)

---

2. 데이터 암호화 시각적 증명 (Visual Proof)

"해커가 데이터베이스 파일을 훔쳐 갔을 때 어떻게 보이는가**?**"

실제 시스템 작동 시 데이터가 암호화되어 저장되는 모습을 비교한 자료입니다.

[상황 **1]** 관리자/앱 화면 **(**정상 접근**)**

관리자가 로그인하여 시스템에 접근했을 때는 복호화된 정상 데이터를 확인할 수 있습니다.

이름: 홍길동 전화번호: 010-1234-5678

[상황 **2]** 데이터베이스 파일 내부 **(**해킹/탈취 시**)**

데이터베이스 파일(`members.db`)을 직접 열어보면, 모든 개인정보가 **난수화(암호화)**되어 있어 내용을 전혀 알아볼 수 없습니다.

ID	Name (이름)	Phone (전화번호)
1	`gAAAAABnKx...` (판독 불가)	`gAAAAABnKx...` (판독 불가)
2	`gAAAAABnKx...` (판독 불가)	`gAAAAABnKx...` (판독 불가)

---

3. 핵심 보안 코드 (Source Code Evidence)

시스템의 심장부인 `app.py`에 적용된 실제 보안 코드입니다.

개인정보 암호화 **(Fernet)**

```
# app.py (Line 39-41)
def encrypt_data(data):
    # 256비트 키를 사용하는 Fernet 암호화 스위트 이용
    return cipher_suite.encrypt(data.encode()).decode()
```

관리자 비밀번호 해싱 **(SHA-256)**

```
# app.py (Line 254)
# 입력받은 비밀번호를 SHA-256 알고리즘으로 단방향 해싱하여 비교
password_hash = hashlib.sha256(password.encode()).hexdigest()
```

Generated by Everest Dev Team | 2026-01-05