Documentation: Implementation Security Properties

1. HMAC for Message Authentication

- **Purpose**: HMAC (Hash-based Message Authentication Code) provides message integrity and authenticity.
- **Security Properties**:
  - **Keyed Hashing**: Uses a secret key to generate a hash, preventing tampering.
  - **Resistance to Collision Attacks**: Based on cryptographic hash functions (e.g., SHA-256).
  - **Replay Attack Prevention**: Can include nonces or timestamps to prevent reuse of valid HMACs.

2. File Integrity Verification System

- **Purpose**: Ensures that a file has not been altered by verifying its HMAC.
- **Security Properties**:
  - **Integrity Verification**: If even one byte changes, the HMAC output changes drastically.
  - **Tamper Resistance**: Any modification to the file requires knowledge of the secret key to generate a valid HMAC.
  - **Secure Storage of Keys**: The security of the HMAC system depends on proper key management. The key should be stored securely and never hardcoded.

Best Practices

- **Use Strong Hash Functions**: SHA-256 or SHA-3 are recommended.
- **Key Management**: Use a secure key derivation function or hardware security module (HSM) for key storage.
- **Salting**: Including a unique identifier or nonce can prevent certain attack vectors.