Project Report: Secure Hybrid File Exchange System
Link to GitHub: https://github.com/einavbs1/Secure_Hybrid_File_Exchange_System
Students: Yuval Lerfeld, Avishag Levi, Einav Momi Ben Shushan

1. Overview & Objectives
This project implements a secure file exchange system designed to guarantee the Confidentiality, Integrity, and Authenticity (CIA) of data. Addressing the trade-off between cryptographic speed and trust, we developed a Hybrid Cryptosystem. This architecture combines the computational efficiency of symmetric encryption for bulk file processing with the robust key management of asymmetric algorithms. The result is a performant and trustworthy solution that solves the key distribution problem inherent in symmetric systems while overcoming the performance limitations of purely asymmetric ones.

2. Cryptographic Methodology
The system integrates three distinct algorithms, each selected for a specific role:

- **Confidentiality (Blowfish in CFB Mode):** We utilize Blowfish, a fast 64-bit block cipher, for file content encryption. By implementing the **Cipher Feedback (CFB)** mode, we transform the block cipher into a self-synchronizing stream cipher. This allows for efficient encryption of files of arbitrary length without the need for data padding.
- **Integrity & Authenticity (Rabin Signatures):** Security is based on the integer factorization problem. We use **Blum Integers** ($p, q \equiv 3 \ (mod\ 4)$) to ensure deterministic root finding. To address the probabilistic nature of Rabin signing, a nonce-based padding scheme is implemented to ensure any file hash can be successfully signed.
- **Key Exchange (EC-ElGamal):** For securely transporting the Blowfish session key, we employ the Elliptic Curve ElGamal protocol over the **secp256k1** curve. This provides high security (equivalent to 3072-bit RSA) with a compact 256-bit key, optimizing bandwidth and performance.

3. System Protocol (Workflow)
The system executes a strict "Sign-then-Encrypt" pipeline to ensure maximum security:

1. **Signing (Alice):** The sender computes the file's hash and generates a Rabin signature using her private key (iterating a nonce until a quadratic residue is found).
2. **Encryption (Alice):** A random Blowfish session key is generated. The payload (File + Signature) is encrypted using Blowfish-CFB.
3. **Encapsulation (Alice):** The session key is encrypted ("encapsulated") using the recipient's (Bob) EC-ElGamal public key.
4. **Decryption & Verification (Bob):** Upon receipt, Bob uses his private EC key to recover the session key, decrypts the file, and validates the enclosed signature against Alice's public Rabin key. This confirms the file originated from Alice and remains intact.

4. Results & Conclusion
The simulation successfully demonstrated the end-to-end protocol. Authenticated users were able to encrypt, sign, transmit, and decrypt files with full validation of data integrity. The system effectively preserves the CIA triad principles. Future enhancements may include upgrading cryptographic primitives to modern standards, such as replacing Blowfish with AES-256 and migrating to Curve25519 for enhanced resistance against side-channel attacks.