From Hybrid Encryption to Transciphering: Lessons from My AES-RSA Implementation

Jaydeep Dineshbhai Roy
ESILV - Master in Cybersecurity & Cloud Computing
jdroy@outlook.in

December 2024

1 Project Context: AES-RSA Hybrid Encryption

In my steganography project, I implemented a hybrid encryption scheme combining symmetric (AES-256) and asymmetric (RSA-2048) cryptography. The architecture was designed to balance security, performance, and practicality:

1. **Data encryption:** AES-256 encrypts the actual message (fast, efficient for large data)
2. **Key protection:** RSA-2048 encrypts the AES key (secure key transmission)
3. **Scheme conversion:** The system converts from symmetric encryption (data) to asymmetric encryption (key), then back to symmetric for decryption

This architecture mirrors real-world protocols like TLS/SSL and is conceptually similar to transciphering in homomorphic encryption.

2 Understanding Overhead: A Practical Lesson

During implementation, I directly experienced the performance and size trade-offs between encryption schemes:

2.1 Performance Overhead

- **AES encryption:** Extremely fast (∼100-200 MB/s on standard hardware)
- **RSA encryption:** Significantly slower (∼1-2 MB/s for the same hardware)
- **Practical impact:** Encrypting even a 256-bit AES key with RSA was noticeably slower than encrypting megabytes of data with AES

2.2 Size Overhead

- **AES ciphertext:** Same size as plaintext (plus small IV/padding)
- **RSA ciphertext:** 256 bytes for just a 32-byte AES key (8x expansion!)
- **Design decision:** This overhead forced me to use RSA only for the small key, not the bulk data

Key Insight: The fundamental trade-off is *security/functionality vs. efficiency.* RSA provides asymmetric key exchange capabilities that AES cannot, but at substantial cost. This trade-off becomes even more extreme with Homomorphic Encryption.

3 Connection to Transciphering in Homomorphic Encryption

After studying recent work on transciphering (particularly AES transciphering for FHE), I recognize striking parallels with my hybrid encryption experience:

3.1 The HE Overhead Problem

Homomorphic Encryption schemes (like BFV, BGV, CKKS) have even more severe overhead than RSA:

- **Ciphertext expansion:** HE ciphertexts can be 1000-10000x larger than plaintexts
- **Computational cost:** Homomorphic operations are orders of magnitude slower than plaintext operations
- **Practical impact:** Transmitting HE-encrypted data from client to cloud is prohibitively expensive

3.2 Transciphering as a Solution

Just as I used AES for bulk encryption and RSA only for key protection, transciphering uses:

1. **Client side:** Fast symmetric encryption (e.g., AES) for data transmission
2. **Server side:** Convert symmetric ciphertext to HE ciphertext *without decryption*
3. **Process:** $HE.Eval(AES.Dec, HE.Enc(k), AES.Enc(m)) = HE.Enc(m)$

This eliminates the need to transmit large HE ciphertexts while still enabling homomorphic computation.

4 Performance Optimization Lessons Learned

1. Minimize Expensive Operations: In my project, I minimized RSA operations by encrypting only the 32-byte key, not the entire message. Similarly, transciphering minimizes HE ciphertext transmission.

2. Algorithm Selection Matters: I experimented with different AES modes (CBC, CTR, GCM). For transciphering, optimizing AES S-box evaluation in the homomorphic domain is equally critical.

3. Measure and Iterate: I measured encryption/decryption times and ciphertext sizes. Transciphering research similarly focuses on measuring homomorphic evaluation depth and noise growth.

4. Real-World Constraints: Theoretical security must be balanced with practical usability. Transciphering embodies this principle.

5 Why I'm Motivated to Work on HE Transciphering

Natural Extension: My hybrid encryption project gave me hands-on understanding of combining cryptographic schemes and optimizing performance trade-offs.

Technical Challenge: Evaluating AES decryption homomorphically requires deep understanding of both AES internals and HE schemes, plus creative optimization.

Real-World Impact: Transciphering enables practical privacy-preserving cloud computing in healthcare, finance, and machine learning.

Research Opportunity: Current implementations still face challenges in computational cost and amortization that I'm excited to help address.

6 Conclusion

My AES-RSA hybrid encryption project provided practical insight into efficiently combining cryptographic schemes with different properties. This experience has prepared me to contribute to cutting-edge research in homomorphic encryption and transciphering. I'm eager to deepen my understanding of lattice-based cryptography and work on optimizing AES transciphering for real-world deployment.