RSA with Provable Primes Using Elliptic Curve Primality Proving

Mạnh Bùi

January 3, 2026

Abstract

This work presents a cryptographic system that integrates the Elliptic Curve Primality Proving (ECPP) algorithm into RSA key generation in order to obtain provable prime factors. Unlike probabilistic primality tests such as Miller–Rabin, ECPP produces a mathematical certificate of primality, increasing trust in high-security cryptographic applications.

1 Introduction

Public-key cryptography relies heavily on the hardness of integer factorization. RSA remains one of the most widely deployed cryptosystems. A critical requirement of RSA is the generation of large prime numbers.

In practice, probabilistic primality tests such as Miller–Rabin are commonly used. While efficient, these tests only provide probabilistic guarantees. Elliptic Curve Primality Proving (ECPP) offers a deterministic alternative by producing a certificate that can be independently verified.

This work implements an RSA cryptosystem whose prime factors are verified using ECPP, thereby providing provable security guarantees.

2 Background

2.1 RSA Cryptosystem

RSA is based on the difficulty of factoring a large composite integer $n = pq$, where p and q are large primes.

2.2 Primality Testing

Probabilistic tests such as Miller–Rabin are fast and widely used, but they do not provide absolute certainty. Deterministic tests are desirable in high-assurance systems.

2.3 Elliptic Curves

Elliptic curves over finite fields provide rich algebraic structures that can be used to design efficient primality proving algorithms.

3 Elliptic Curve Primality Proving

ECPP is a deterministic algorithm that proves the primality of an integer by constructing elliptic curves over finite fields and verifying group order properties.

Given an integer n, ECPP attempts to find an elliptic curve $E/\mathbb{Z}_n$ such that the order of the group $E(\mathbb{Z}_n)$ satisfies certain factorization properties. If these properties hold and a large prime factor is recursively proven prime, then n is prime.

Unlike probabilistic tests, ECPP produces a certificate that can be independently verified.

4 System Design

The system is divided into two main components:

- An ECPP-based primality prover and verifier
- An RSA cryptosystem that uses ECPP-verified primes

Figure 1: System architecture integrating ECPP and RSA

5 RSA with Provable Primes

Prime numbers are first generated using Miller–Rabin for efficiency. Each candidate prime is then verified using ECPP to obtain a primality certificate.

The RSA modulus $n = pq$ is constructed using these verified primes. The certificates can be stored alongside the private key and used for independent verification.

6 Experiments

Experiments were conducted to evaluate the correctness and performance of the system.

6.1 Correctness

All generated RSA keys successfully passed encryption, decryption, and digital signature tests.

6.2 Performance

While ECPP is significantly slower than Miller–Rabin, it is used only as an offline verification step. The overall overhead is acceptable for high-security applications.

7 Conclusion

This work demonstrates the feasibility of integrating ECPP into RSA key generation to obtain provable primes. The resulting system provides stronger security guarantees than conventional RSA implementations.

Future work includes optimizing ECPP for large integers, implementing full Atkin–Morain ECPP, and extending the system to other cryptographic primitives.