ML-based Attack on Digitally Authenticated RSA Algorithm using Model Estimation

Aurelius Nguyen

March 17, 2025

1 Introduction

The RSA cryptosystem remains a cornerstone of modern cryptography, widely used for secure communications, digital signatures, and key exchange protocols. Its security is fundamentally based on the computational difficulty of factoring large semiprimes—numbers that are the product of two large prime numbers—which makes the system robust against attacks using classical computing methods [7]. Traditional factorization methods, such as the General Number Field Sieve (GNFS), require sub-exponential complexity to factor large RSA moduli, thus ensuring that RSA encryption remains secure against current classical computational threats [7]. However, recent advances in machine learning (ML) have demonstrated surprising capabilities in cryptanalysis, particularly in recognizing hidden patterns in large datasets. Preliminary research suggests that deep learning models may be able to learn probabilistic properties of primes and semiprimes, potentially aiding in the prediction of factors more efficiently than brute-force methods [6]. This research explores whether ML-based models can effectively factor large semiprimes, thereby exposing potential vulnerabilities in RSA implementations. If successful, such an approach could significantly impact the cryptographic community, urging faster transitions toward post-quantum cryptographic standards [2].

2 Background

RSA (Rivest-Shamir-Adleman) is a public-key cryptosystem that underpins many secure communication protocols today. It operates on the principle that, while it is computationally simple to multiply two large prime numbers together to form a semiprime, it is exceedingly difficult to reverse the process by factoring the resulting composite number. This one-way function is what provides RSA with its security. The standard method for attacking RSA is by factorizing its large modulus, a task that becomes impractical as the key size increases. For instance, the General Number Field Sieve (GNFS) is currently the most efficient classical algorithm for factoring large integers, yet its sub-exponential time complexity ensures that RSA remains secure for sufficiently large key sizes [7]. Additionally, prior research has shown that neural networks can predict small prime factors with reasonable accuracy on lower-bit numbers [6], suggesting that ML might offer alternative insights into the factorization problem. RSA is also widely used in digital authentication processes, further highlighting the importance of ensuring its security [7].

Neural network architectures have been around for a while. However, the work by Nene and Uludag [6] primarily employs feed-forward LSTM-RNN-based neural networks in a binary framework for prime factor prediction. What is new in our approach is the integration of modern mechanisms such as transformer-based models and Generative Adversarial Networks (GANs) and. These advancements leverage huge compute capacity and adversarial training strategies—in which a generator refines candidate primes and a discriminator helps steer predictions closer to true prime values. Additionally, our approach incorporates sophisticated feature engineering that utilizes mathematical properties (e.g., modular residues, Hamming weights, and ECPP-based signatures) to provide a richer representation of semiprime structures.

3 Hypothesis

We hypothesize that machine learning models can extract underlying patterns in semiprimes, enabling a predictive approach to integer factorization that surpasses brute-force and traditional techniques. In particular, we anticipate that neural network architectures will be able to learn structural properties of prime numbers and their compositions within semiprimes. Feature extraction techniques, such as modular residues and ECPP-based signatures, are expected to enhance model accuracy. Moreover, we propose that Generative Adversarial Networks (GANs) may improve the efficiency of prime detection through an adversarial setup, where a generator iteratively refines candidate primes while a discriminator distinguishes true primes from near-prime candidates.

To clarify the overall flow of our approach: we begin by sourcing high-quality prime and semiprime datasets from established online repositories. Next, robust feature engineering techniques are applied to extract numerical characteristics that capture intrinsic mathematical properties. These engineered features are then input to various ML models—including LSTM, transformer-based, hybrid architectures, and GANs—each designed to learn and predict the factorization of semiprimes. The models are trained using distributed computing resources, and their performance is evaluated against classical factorization methods. Ultimately, our expectation is that the proposed ML models will demonstrate measurable performance improvements over traditional approaches when applied to semiprimes of realistic RSA key sizes [6].

4 Methodology

4.1 Data Sources

Instead of generating prime and semiprime numbers manually, we can utilize existing online databases that store precomputed values:

- **PrimePages**: Maintains a comprehensive database of prime numbers, including the 5000 largest known primes. Available at: `https://t5k.org/`
- **Prime Numbers List**: Provides a list of prime numbers up to 1 trillion. Available at: `https://compoasso.free.fr/primelistweb/page/prime/liste_online_en.php`
- **Prime-Numbers.info**: Offers lists of semiprime numbers up to 1000. Available at: `https://prime-numbers.info/list/semiprimes-up-to-1000`

- **OEIS (Online Encyclopedia of Integer Sequences)**: Contains sequences related to semiprimes, such as sequence A001358, which lists numbers that are the product of two primes. Available at: `https://oeis.org/A001358`

4.2 Feature Engineering

- **Binary encoding**: $N \to (b_0, \ldots, b_{k-1}) \in \{0,1\}^k$
- **Modular residues**: $N \mod m$ for $m \in \{3, 5, 7, 11, 13\}$
- **Hamming weight**: $H(N) = \sum_{i=0}^{k-1} b_i$
- **ECPP-based prime signatures**: Utilizing the elliptic curve primality proving (ECPP) method to extract prime-related characteristics that may serve as additional features for ML models [1].

4.3 Model Architecture

- Transformer-based models and LSTM leveraging attention mechanisms
- Hybrid models combining convolutional and recurrent components
- Generative Adversarial Networks (GANs)

4.4 Training Protocol

- 80-10-10 train/validation/test split
- Dual loss for p and q prediction:

$$\mathcal{L} = -\frac{1}{2k} \sum_{i=0}^{k-1} \left[y_i^{(p)} \log \hat{y}_i^{(p)} + y_i^{(q)} \log \hat{y}_i^{(q)} \right]$$

5 Significance

A successful ML-based approach to RSA factorization could challenge the computational security of RSA by revealing previously unidentified structural weaknesses in semiprimes. This method could provide alternative insights into the integer factorization problem, potentially reducing the effective bit length required for secure RSA keys. Such advancements may not only accelerate cryptanalysis efforts by complementing traditional factorization methods like the GNFS [7] but also drive a faster adoption of post-quantum cryptographic standards by highlighting vulnerabilities in current encryption practices [2]. The implications extend beyond theoretical research, as practical breakthroughs in ML-based factorization might necessitate urgent changes in cryptographic policies and promote the rapid deployment of lattice-based or quantum-resistant cryptosystems.

6 Faculty Mentor

Professor Ali Anwar, who specializes in Distributed Systems & ML, will supervise weekly progress reviews and methodology validation.

References

[1] Atkin, A. O. L., & Morain, F. (1993). Elliptic curves and primality proving. *Mathematics of Computation*, 61(203), 29-68. DOI: 10.1090/S0025-5718-1993-1199989-X.

[2] Barker, E., & Dang, Q. (2015). *Recommendation for key management: Part 3 – Application-specific key management guidance*. NIST Special Publication 800-57.

[3] Hellman, M. E. (1979). The mathematics of public-key cryptography. *Scientific American*, 241(2), 146-157.

[4] Jansen, K. N. B. (2005). Neural networks following a binary approach applied to the integer prime-factorization problem. *2005 IEEE International Joint Conference on Neural Networks*.

[5] Murat, B., Kadyrov, S., & Tabarek, R. (2020). Integer prime factorization with deep learning. *Journal of Cryptographic Engineering*, 10(3), 201-215.

[6] Nene, R., & Uludag, S. (2022). Machine learning approach to integer prime factorisation. *Journal of Cryptology*, 39(4), 1-24.

[7] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2), 120-126.