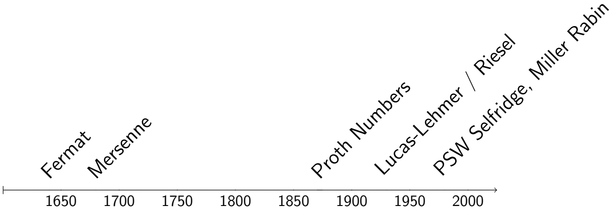
Prime Heuristics and their Implications on RSA Cryptography

Sitar Eswar

July 12, 2024

Introduction

- Importance of Prime Heuristics:
  - Prime numbers have been researched for millennia as one of the most fundamental aspects in number theory.



- Comparison of Prime Heuristics:
  - We evaluate time efficiency, computational effectiveness, and overall performance.

Goals and Applications of Prime Heuristics

- Diverse Applications:
  - Prime heuristics have diverse appications in the fields of cryptography, error correction codes, and cryptocurrencies.
- Focus on RSA Cryptography:
  - Prime heuristics play a pivotal role in selecting secure prime numbers for RSA encryption.
- Goal:
  - Identify the most optimal heuristic for RSA encryption and decryption as well as its resilience to brute-force attacks.

Fermat Primality Test

- Is a probabalistic algorithm that checks if a number n is prime by verifying that $a^{n-1} \equiv 1 \mod n$. Efficient, but it misidentifies Carmichael numbers as prime as well.
- **Pros:** Correctly identifies all primes. The time required to perform the computations is very effecient, taking only 1 second to print primes from 1 through 100,000.
- **Cons:** Has a high False-Positive rate of around 150 in the first 100,000 numbers.
- Carmichael numbers are composite numbers that pass Fermat's primality test for most bases, making them counterexamples that can falsely appear prime under this test.

Miller-Rabin Primality Test

- Checks whether a specific property of primes holds for the number that is being tested. Seperate a number n into $2^a * b + 1$, where $a \geq 1$, and b is odd.
- Pick a random $y \in \{1, 2, \ldots, n-1\}$. If $y^b \equiv \pm 1 \mod n$, then n is prime. If $y^{2^r * b} \equiv -1 \mod n$, where $0 \leq r \leq a-1$, n is prime. If this is also false, then n is composite.
- **Pros:** Very reliable in identifying primes and composites, if given sufficient number of witnesses. It is almost as reliable as the Fermat test.
- **Cons:** Largely depends on what the witness (number generated at random) is.. i.e, 9 may be marked as prime if 8 is a witness. The time necessary to perform the computations is more than Fermats.

Proth Primality Test

- Used to determine whether a Proth number is prime. A Proth number is of the form $k \cdot 2^n + 1$, where k is an odd integer and $2^n > k$. The test states that a Proth number p is prime if there exists an integer a such that $a^{(p-1)/2} \equiv -1 \pmod{p}$.
- **Pros:** Very time-efficient, taking only 0.9 seconds to compute all Proth primes from 1 through 100,000.
- **Cons:** Has a high False-Positive rate. Does not produce as many primes as Fermat, Miller-Rabin, and PSW-Selfridge. It is only applicable to proth numbers.

Mersenne Primes and Lucas-Lehmer Test

- Mersenne Primes are a special class of prime numbers in the form of $M = 2^n - 1$, where n itself is a prime number.
- Method to verify whether a Mersenne Number is prime. Begins by constructing a sequence, s_i, such that $s_0 = 4$, and $s_i = (s_{i-1}^2 - 2)$, for subsequent s_i's. According to the Lucas Lehmer Test, M is prime if and only if s_{n-2} is congruent to 0 mod M.
- **Pros:** Does not fail; there is lots of leniency in selecting the function/sequence for the prime check.
- **Cons:** Computationally ineffective. It is not time-effecient.

Lucas-Lehmer-Riesel Test

- A method to verify whether a certain number of the form $M = k * 2^n - 1$ is prime. Like the Lucas-Lehmer Check, uses a sequence to prove the primality of M.
- Construct a sequence s_i such that $s_i = (s_{i-1}^2 - 2)$. The values of k depends on our value of s_0. If $k = 3$ and $n \equiv 0$ or $3 \pmod 4$, we can take $s_0 = 5778$. However, if $k \equiv 1 \mod 6$ and n is odd, or $k \equiv 5 \mod 6$ and n is even, $M \equiv 7 \mod 24$. If this is the case, we take $s_0 = (2 + \sqrt{3})^k + (2 - \sqrt{3})^k$.
- **Pros:** Does not fail in identifying whether the testing number is prime or composite.
- **Cons:** Computationally ineffecient. Both the primality check and the time required to select a starting value s_0 of the sequence are ineffecient.

Selfridge, Pomerance, Wagstaff Heuristic

- If p is an odd number, p is prime if the following hold:

$$p \equiv \pm 2 \mod 5$$
$$2^{p-1} \equiv 1 \mod p$$
$$f_{p+1} \equiv 0 \mod p$$

  where f_n denotes the n^{th} fibonacci number.
- **Pros:** PSW has 0 False-Positives (up until 100,000), meaning that no numbers that are composite are marked as prime by the heuristic.
- **Cons:** Demonstrates a significantly low time-efficiency rate, at 28 seconds. PSW is only applicable to numbers of the form $\pm 2 \mod 5$.

Overview of RSA Public-Key Cryptography

- RSA (Rivest-Shamir-Adleman) is a widely used public-key cryptosystem for secure data transmission.
- It is based on the mathematical fact that factoring the product of two large prime numbers is difficult.
- RSA uses a pair of keys: a public key for encryption and a private key for decryption.
- The public key is shared with everyone, while the private key is kept secret.

RSA Key Generation Process (Example): Alice

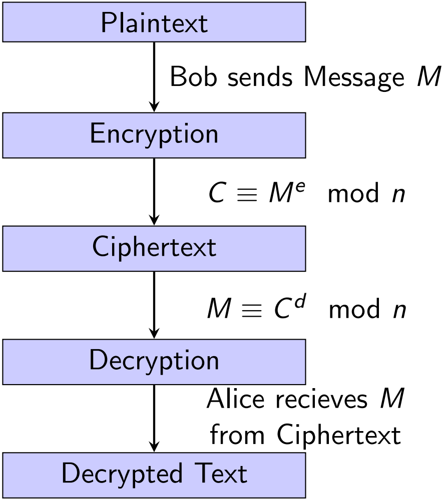
- Alice, the host, first chooses two large prime numbers: 61 and 53, as part of her private key.
- She calculates $n = 61 \cdot 53 = 3233$, a part of her public key.
- After this, she computes $\lambda(n)$ using Carmichael's totient function:

$$\lambda(n) = \mathrm{lcm}(60, 52) = 780$$

- She then selects e (public exponent) coprime to $\lambda(n)$: $e = 17$.
- Computes d (private exponent) as the modular multiplicative inverse of e modulo $\lambda(n)$: $d = 413$.
- Hence, Alice's public key is $(17, 3233)$, and her private key is $(413, 61, 53, 780)$.

RSA Message Exchange Scenario (Example): Bob

- Bob must first convert the message to a numerical value M: $M = 11$.
- Bob encrypts M using Alice's public key:
- He calculates the ciphertext C using modular exponentiation: $C \equiv 11^{17} \mod 3233 = 3061$.
- Bob then sends the ciphertext C to Alice.



Decryption Process

- Alice calculates M as $M \equiv C^d \mod n$.
- Example calculation: $M \equiv 3061^{413} \mod 3233 \equiv 11$.
- Alice interprets M back into the original message Bob encrypted.
- Only Alice, with her private key, can decrypt the message that Bob encrypted with her public key, ensuring secure communication.

Attacks on RSA Cryptography

- Hastad's Broadcast Attack (1988): Exploits small private exponents by intercepting multiple ciphertexts to recover the private key using (CRT).
- Goldberg and Wagner's PRNG Attack 1 (1996): Predicts Netscape's encryption key using system process IDs (pid, ppid) and time of challenge transmission.
- Goldberg and Wagner's PRNG Attack 2 (1996): Efficient brute-force attack on Netscape's encryption by approximating predictable pid and ppid's.
- Opportunistic Mining of "P's" and "Q's": Heninger et al. take advantage of weak random number generators and GCD attacks to find private keys by harvesting public keys.

Prime Number Density

- The Prime Number Theorem states that the number of primes less than a given number x approximates $\frac{x}{\log x}$.
- The second Hardy-Littlewood conjecture proposes the asymptotic density of the number of primes in an interval.

$$\pi(x+y) - \pi(x) \leq \pi(y)$$

- We analyze the distribution and density of prime numbers within the ranges of 32-bit, 64-bit, 128-bit, and 256-bit integers. Our results are in graphs below.
- Compare the observed densities and distributions with the theoretical predictions from the Prime Number Theorem and Hardy-Littlewood conjecture, assessing the accuracy and any deviations.

Efficacy of Prime Heuristics

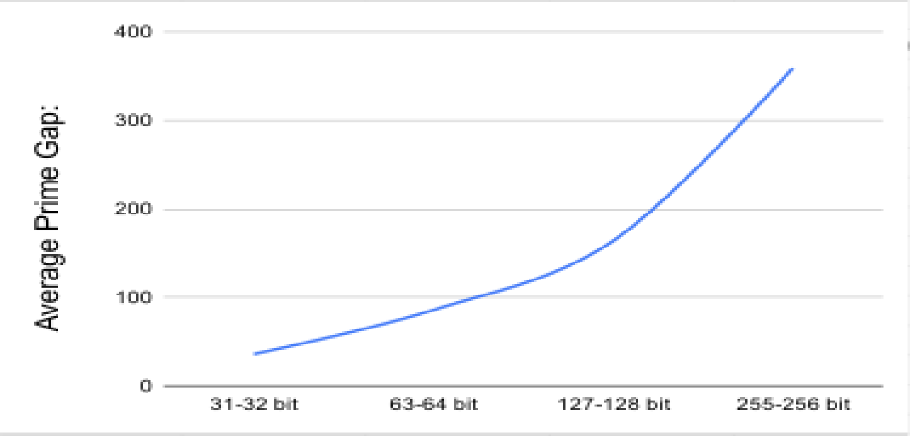
- Assessed the efficacy of various prime computation methods for number theory, RSA cryptography, and computational mathematics, using prime density as a model for evaluating prime heuristics.

- Generated random odd numbers at 255-256 bit scale, tested primality with Miller-Rabin and Fermat tests, calculated gaps between consecutive primes, and repeated for 32 bit, 64 bit, and 128 bit numbers. Graph will be displayed in further slides.

Efficacy of Prime Heuristics

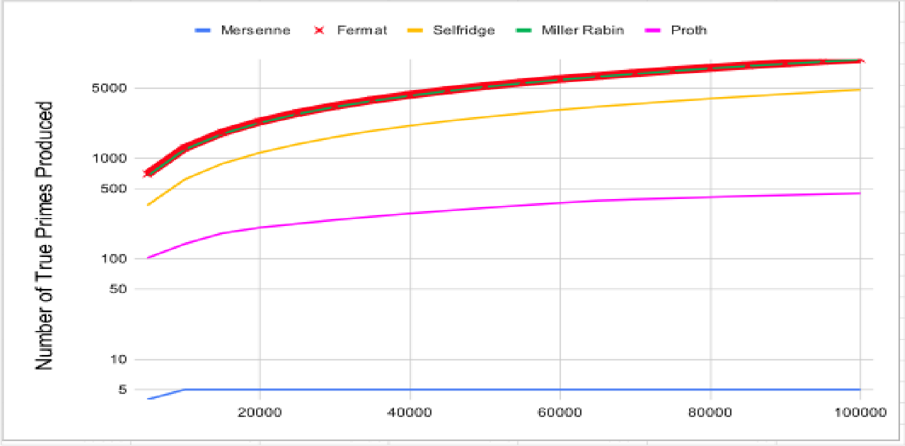
- Evaluated Mersenne, Fermat, Selfridge, Miller-Rabin, and Proth based on the quantity of true primes generated, production time efficiency, and incidence of false positives.
- Mersenne and Proth are inefficient for generating primes below 100,000, Fermat, Miller-Rabin, and Selfridge are the most effective.
- Miller-Rabin and Fermat's are the most time efficient. PSW-Selfridge took the longest.
- Concurred that Miller-Rabin is the most effective prime heuristic.

Prime Gaps in Large Bit Values



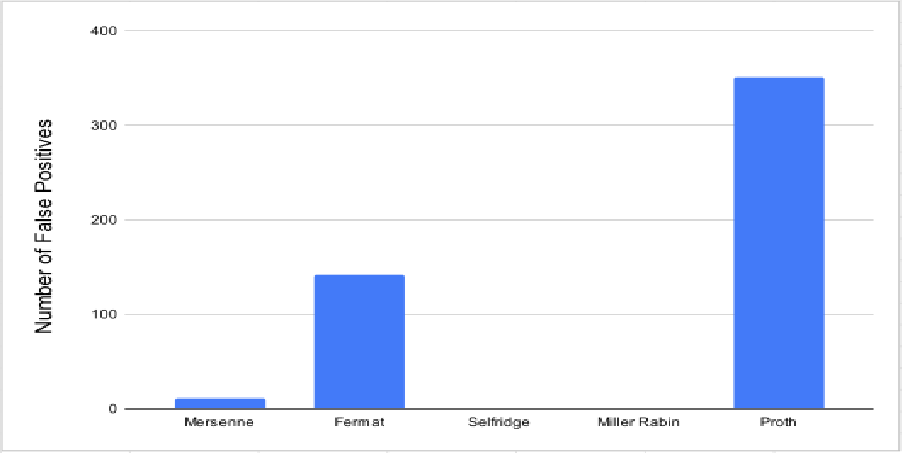
Average prime gap increases as bit length increases, closely following the prime number theorem

Number of Correctly Identified Primes from 1 through 100,000



Number of primes produced by each Prime Heuristic

False Positive Rates of Prime Heuristics



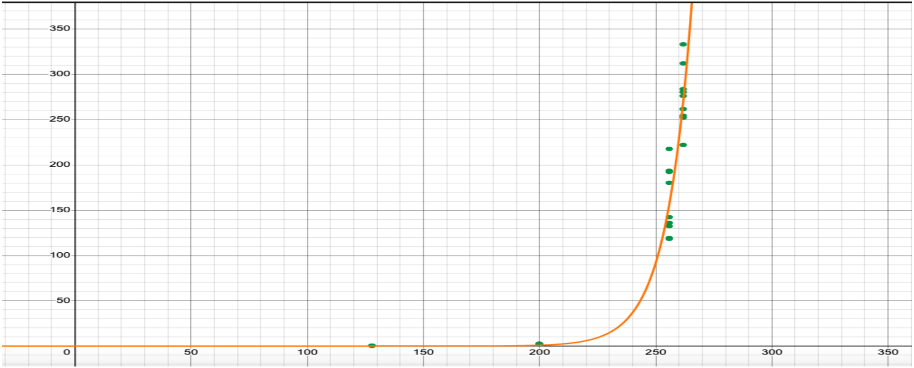
Number of False Positives throughout the 5 heuristics

Using MSieve to Factors RSA Moduli

- Number Field Sieves
  - Number field sieves are used for factoring extremely large numbers that traditional methods cannot handle efficiently.
- MSieve
  - MSieve is a C library for integer factorization that utilizes both the quadratic sieve (QS) and the number field sieve.
  - It is optimized for large composite integers, including RSA moduli, by efficiently finding their prime factors.
  - MSieve is particularly effective for factoring moduli in the range of 256 bits and larger.
  - Below is a graph displaying the time in seconds to factor a number (bit length is listed) made by multiplying two large prime numbers.

Time required to factor moduli of various bit length using number sieves.

x-axis: number of bits; y-axis: time in seconds.



Experiments were run using the Msieve library on a 4-CPU 2.9 Ghz Ubuntu system. Approximately 3 mins to factor 256-bit number, but computation time rises exponentially!

Discussion

- Prime heuristics will help identify prime numbers efficiently, which is crucial for generating RSA keys and also for attacking RSA by finding plausible prime factors of the modulus n.
- Targeting the correct range for potential prime factors is important as it will narrow down the number of plausible prime factors for the modulus. Note: n bit composite numbers' factors are usually $\frac{n}{2}$bits.
- Utilization of prime heuristics helps in the identification of primes and composites. Specifically, the Fermat and Miller-Rabin primality tests are the most effective strategies in recognizing primes.
- Number Sieve implementations greatly outperform naïve approaches to brute-forcing RSA. 1024-bit RSA cryptography seems sufficiently robust against contemporary attacks.