Number Theory And Cryptography CO313 Paper Implementation Mini Project

Siddesh L C – 16CO144

Shreyas Pandith – 16CO142

Paper Selected for Mini Project

Fast Prime Generation Algorithms using proposed GCD test on Mobile Smart Devices

https://ieeexplore.ieee.org/document/7425951

- A prime is a natural number that is bigger than 1 and has no positive divisors except 1 and itself.
- N-bit Prime generation steps generally
  - A n-bit positive odd random number r
  - The primality test examines whether r is a prime or not.
- Th important task is developing fast primality tests.
- A deterministic primality test
  - certifies that r is a prime with probability 1
  - GCD test, elliptic curve analogue and Maurer’s algorithm.
- A probabilistic primality test
  - certifies that r is a prime with high probabilities closer to 1
  - Fermat test, Miller-Rabin test and Solovay-Strassen test.

Popular primality tests

- Trial division and Miller-Rabin test (TD-MR combination hereafter)

- GCD test and Miller-Rabin test(GCD-MR combination hereafter)

Fermat’s Theorem

- Fermat’s Theorem
  - if n is prime, then for any a we have

$$a^{(n-1)} \equiv 1 \pmod{n}.$$

  - $a \in \{1,\ldots,n-1\}$

  - If not, then n must be composite

The Miller-Rabin Test

- Suppose n is prime with $n > 2$ and It follows that $n - 1$ is even and we can write it as (2^s)*d , where s and d are positive integers (d is odd).

  - => $a \wedge d \equiv 1 \pmod{n}$

  - OR => $a \wedge ((2 \wedge r) * d) \equiv -1 \pmod{n}$, $0 \le r \le s - 1$

The Miller-Rabin Test

- In practice, we implement the Miller-Rabin test as follows:
  1. Given n, find s so that $n-1=(2^s)*d$ for some odd d.
  2. Pick a random $a \in \{1,...,n-1\}$
  3. If $a \wedge d \equiv 1 \pmod{n}$ then n passes (and exit).
  4. For $i = 0,...,d-1$, see if $a^{((2^i)*d)} \equiv -1 \pmod{n}$. If so, n passes (and exit).
  5. Otherwise n is composite.

TD-MR Combination(n, k)

1. Random Number Generation
   - Generate an n-bit odd random number r.
2. Trial division on r with k primes
   - Divides r by k small primes.
   - If r is divided by any prime, go to Step 1.
3. Miller-Rabin test on r
   - Perform Miller-Rabin Test on r.
   - If r passes, return r as a prime.
   - Otherwise, go to Step 1.

k small primes => k primes less than or equal to $\sqrt{n}$

GCD-MR combination(n, k)

1. Random Number Generation
   - Generate an n-bit odd random number r.
2. GCD test on r with k primes
   - Find GCD of r and k small primes individually.
   - If GCD is not 1 for any prime with r go to Step 1.
3. Miller-Rabin test on r
   - Perform Miller-Rabin Test on r.
   - If r passes, return r as a prime.
   - Otherwise, go to Step 1.

k small primes => k primes less than or equal to $\sqrt{n}$

Proposed new Algorithms by Authors

- Since the running time of GCD test is slower than the running time of division, if the number of GCD test and the number of division are the same, GCD-MR combination is always slower.
- PGCD-MR combination
- MGCD-MR combination.

PGCD-MR Combination

- GCD(r, p1 · p2)=1

  $\Leftrightarrow$ GCD(r, p1)=1 and GCD(r, p2) = 1)

- We define PGCD that computes the greatest common divisor between r and Πk where Πk is the product of small primes.

  PGCD(r, k) = GCD(r, Πk)

PGCD-MR Combination(n, k)

1. Random Number Generation
    - Generate an n-bit odd random number r.
2. GCD test on r and Πk
    - Computes GCD(r, Πk).
    - If the result is not 1, go to Step 1.
3. Miller-Rabin test on r
    - Perform Miller-Rabin Test on r.
    - If r passes, return r as a prime.
    - Otherwise, go to Step 1.

Πk => product k primes less than or equal to √n

MGCD-MR Combination

- PGCD becomes slower as the bit-length of Πk becomes bigger and bigger.
- The key idea of MGCD is dividing Πk into the several proper bit-length of Πkj .
- MGCD(r, Πk)=1

  $\Leftrightarrow$ GCD(r, Πk1)=1, GCD(r, Πk2)=1, ... GCD(r, Πks)=1
- MGCD computes the greatest common divisor between r and Πki sequentially until finding the gcd of r and Πki is not one.

MGCD-MR Combination(n, k)

1. Random Number Generation
   - Generate an n-bit odd random number r.
2. GCD test on r and Πkj
   - Divide Πk into the proper length of Πk.
   - Computes GCD(r, Πkj) sequentially
   - If the result is not 1, go to Step 1.
3. Miller-Rabin test on r
   - Perform Miller-Rabin Test on r.
   - If r passes, return r as a prime.
   - Otherwise, go to Step 1.

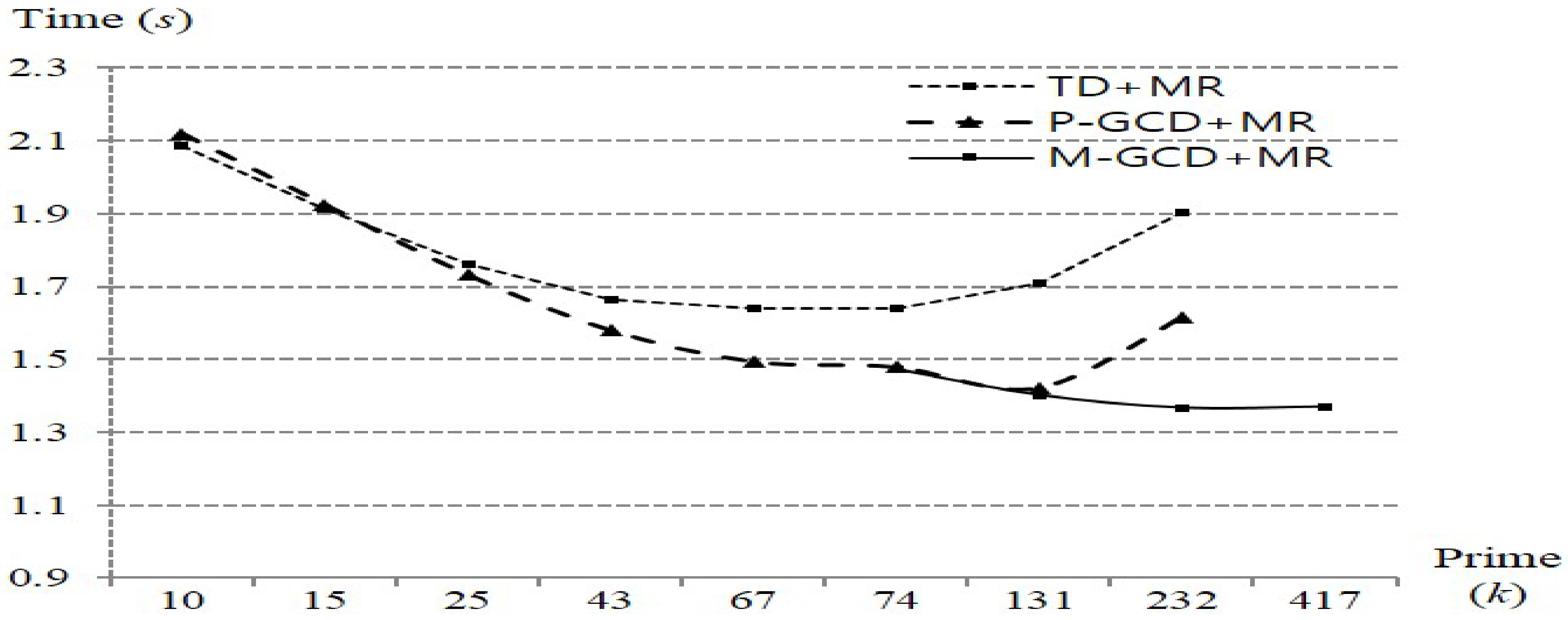
Πkj => product j < k primes less than or equal to √n

Total running time

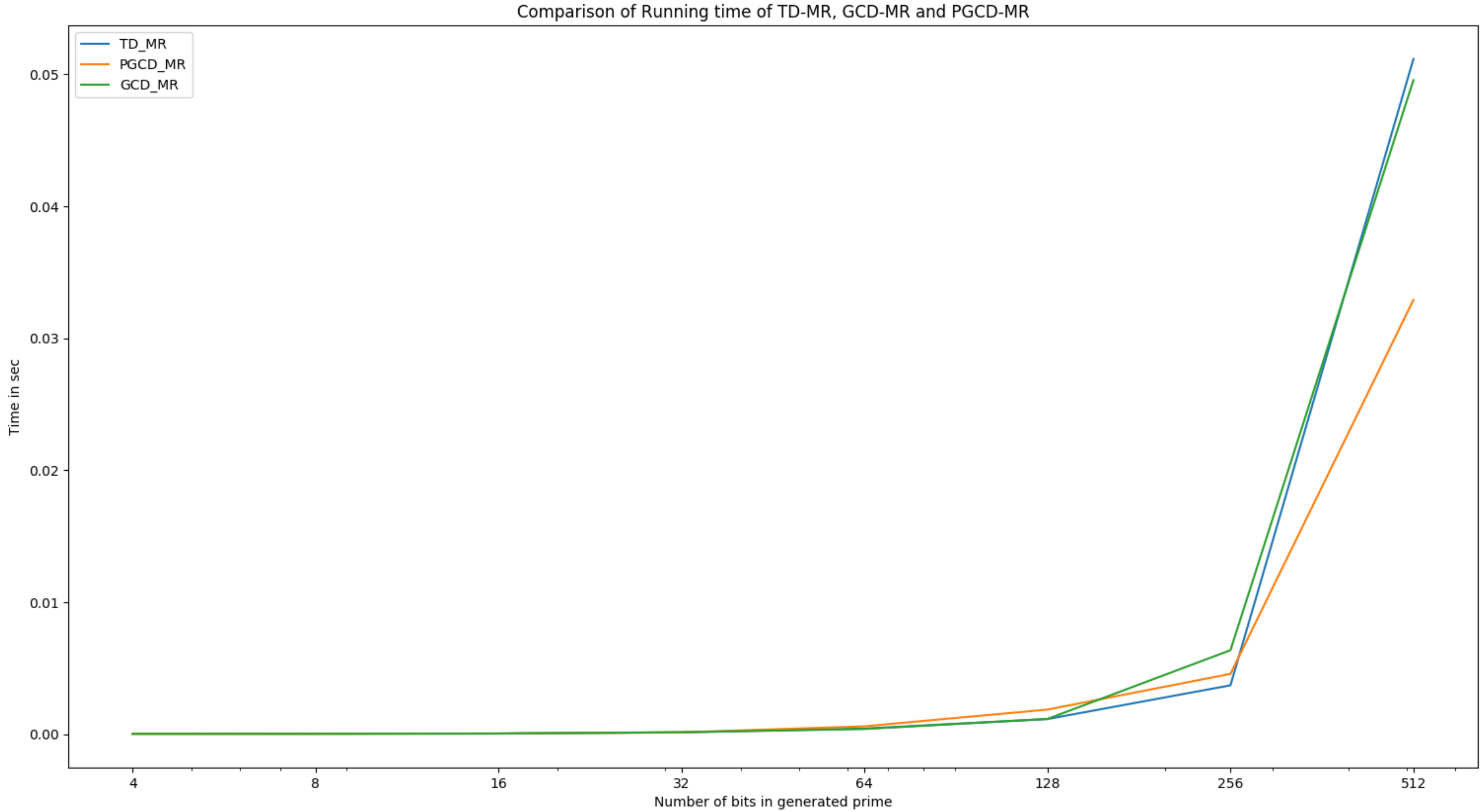
- TD-MR Combination
  - $T = NT \cdot (TRND + TTD + TMR)$
- GCD-MR Combination
  - $T = NT \cdot (TRND + TGCD + TMR)$
- PGCD-MR Combination
  - $T = NT \cdot (TRND + TPGCD + TMR)$
- MGCD-MR Combination
  - $T = NT \cdot (TRND + TMGCD + TMR)$

Result Comparison

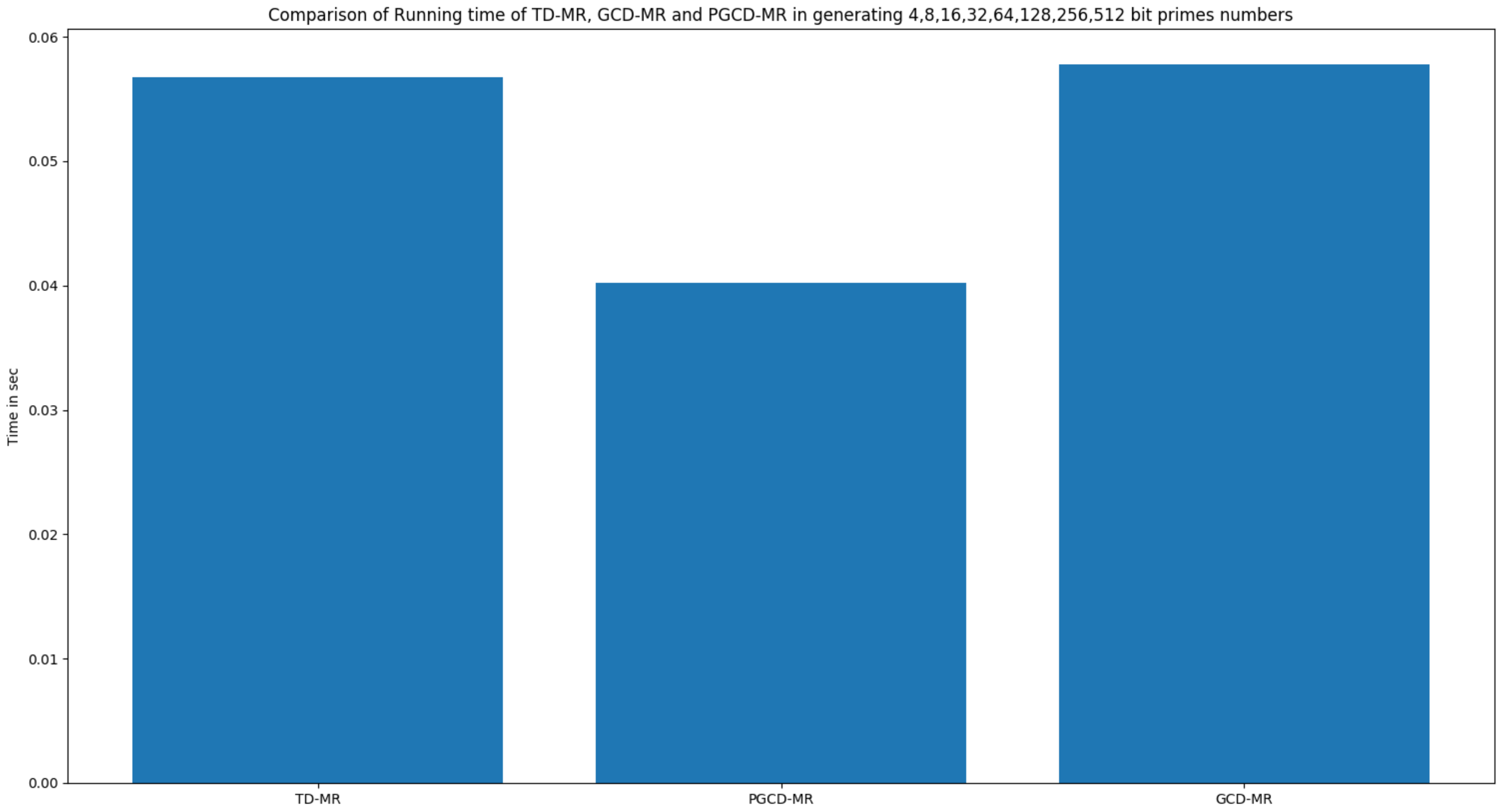
- The result shows the running time of TD-MR, PGCD-MR and MGCD-MR combination when 1,024 bit prime generated.



Our Work Till Date



Our Work Till Date



Thank You