1 Prime Number Generation

1.1 Uniform prime generation

The original 3-step algorithm [cite], in essence, generates two random prime numbers that are distributed uniformly in a given interval. Ahlswede then uses the fact that the primes are uniform to bound the second kind error probability. [The relaxation of this assumption complicates the error analysis. Explore this in subsection x.].

To generate uniform primes on the interval $]1, \pi_K]$, Ahlswede picks a random index $k \leftarrow \{1, \ldots, K\}$ and then calculates π_k. Although this method minimizes the number random bits and generates exactly uniform primes, it is computationally expensive with state-of-the-art algorithms compute π_k in $2^{\mathcal{O}(K)}$[cite].

To improve the time complexity of the scheme, we look for polynomial time algorithms that produce almost uniform primes with the least number of random bits possible. A trivial algorithms for producing uniform primes is given in Algorithm 1.

Algorithm 1: Generating uniform primes

input : Positive integer n.
output: A uniformly choen prime number less than or equal to n.
1 **repeat**
2     $p \leftarrow \{2, \ldots, n\}$;
3 **until** *p is prime*
4 **return** p

When we use a deterministic primality test in line 3 of Algorithm 1 the distribution of primes is exactly uniform. This algorithm may never terminate, however, we expect it to stop after $\mathcal{O}(\lg n)$ steps. Because, by the prime number theorem [reference],

$$\frac{\pi(n)}{n} \approx \frac{1}{\ln n} \tag{1}$$

hence, on average after $\mathcal{O}(\lg n)$ steps a prime number p is chosen. As a result, Algorithm 1 uses an average of $(\lg n)^2$ random bits. The current state-of-the-art deterministic primality tests, AKS, runs in $\tilde{\mathcal{O}}\Big((\lg n)^6\Big)$ which means that on average Algorithm 1 runs in $\tilde{\mathcal{O}}\Big((\lg n)^7\Big)$ [cite].

We can further improve the time complexity of the Algorithm 1 if we use randomized primality tests. These tests can determine whether a number p is prime with high probability.

Algorithm 2: Generating uniform primes

input : Positive integer n.
output: A uniformly choen prime number less than or equal to n.
repeat
    $p \leftarrow \{2, \ldots, n\}$;
until *p is probably a prime*
return p

For example, the Miller-Rabin test might declare a composite number as a prime, however, this happens with low probability, as low as desired. The output of the Algorithm 2 is not a unifrom prime number as it can be composite, however, the distribution of prime numbers is equiprobable over all primes. Continuing with Miller-Rabin test, it uses $\lg p$ random bits where p is the number that is to be tested. Therefore, we still use an average of $2(\lg n)^2$ random bits. The test itself runs in $\mathcal{O}\Big((\lg n)^3\Big)$ hence, the Algorithm 2 runs in $\mathcal{O}\Big((\lg n)^4\Big)$.

In this report, we implement the Miller-Rabin test since it is more efficient and easier to implement. Furthermore, by exectuing this test an appropriate number rounds, we can ensure that the resulting distribution is statistically close to the uniform distribution over primes.

1.2 Miller-Rabin analysis

Miller-Rabin is a well-known random primality test algorithm. Let $MR(n, k)$ be the distribution of Miller-Rabin algorithm on the prime candidate n with k repeats. Let $\mathcal{P}$ be the set of primes. Then, from Theorem 31.39 of [4] we have the following probabilities.

$$\Pr[MR(n,k) \mid n \in \mathcal{P}] = 1 \tag{2}$$
$$\Pr[MR(n,k) \mid n \notin \mathcal{P}] \leq 4^{-k} \tag{3}$$

Consider the following random prime number genrator, $GMR(N, s, k)$. This algorithm, samples numbers uniformly and then checks if they are prime using the Miller-Rabin test. The parameter N is the upperbound, s is the maximum number of attempts, and k is the number of repeats in the underlying Miller-Rabin test. We analyse

Algorithm 3: $GMR(N, s, k)$

```
input  : positive integers N, s, k
output: A uniformly chosen prime number less than or equal to N
for i = 1 → s do
    n ← {1, 2, ..., N}
    if MR(n, k) then
        return n
    end
end
return ⊥
```

the distribution of $GMR(N, s, k)$. Let n_i denote the random variable n in the i_{th} iteration.

$$\begin{aligned}
\Pr[GMR(M, s, k) = \perp] &= \Pr[MR(n_1, k) = \cdots = MR(n_s, k) = 0] && (4)\\
&= \prod_{i=1}^{s} \Pr[MR(n_i, k) = 0] && \text{(Independence)} \quad (5)\\
&= \prod_{i=1}^{s} \Pr[MR(n_i, k) = 0 \mid n_i \notin P] \Pr[n_i \notin P] && (6)\\
&\leq \prod_{i=1}^{s} \left(1 - \frac{\pi(N)}{N}\right) && (7)\\
&= \left(1 - \frac{\pi(N)}{N}\right)^{s} && (8)\\
&\approx \left(1 - \frac{1}{\ln N}\right)^{s} && \text{(PNT)} \quad (9)
\end{aligned}$$

If we bound this error probability with ϵ, then we get the following bound on s.

$$s \geq \frac{\ln \epsilon}{\ln\left(1 - \frac{1}{\ln N}\right)} \approx -\ln N \ln \epsilon$$

for sufficiently large N.

The probability that the result of $GMR(N, s, k)$ is composite, given it is not $\perp$ is as follows.

$$\begin{aligned}
\Pr[GMR(N, s, k) \notin P \mid GMR(N, s, k) \neq \perp] &\leq \sum_{i=1}^{s} \Pr[MR(n_i, k) = 1, n_i \notin P] && (10)\\
&= \sum_{i=1}^{s} \Pr[MR(n_i, k) = 1 \mid n_i \notin P] \Pr[n_i \notin P] && (11)\\
&= \sum_{i=1}^{s} 4^{-k} \left(1 - \frac{\pi(N)}{N}\right) && (12)\\
&= s4^{-k} \left(1 - \frac{\pi(N)}{N}\right) && (13)\\
&\approx s4^{-k} \left(1 - \frac{1}{\ln N}\right) && \text{(PNT)} \quad (14)
\end{aligned}$$

If we bound this error probability with δ, then we get the following bound on s.

$$s \leq \left(1 - \frac{1}{\ln N}\right)^{-1} 4^k \delta \approx 4^k \delta \tag{15}$$

Let $\epsilon = e^{-l}$ and $\delta = 2^{-q}$ with $l, q \geq 0$. Then,

$$l \ln N \leq s \leq 2^{2k-q} \tag{16}$$

Note that, $s = l \ln N$ and $k = \dfrac{\lg(l/e) + \lg\lg N + q}{2}$ satisfies both inequalities.

1.3 Prime Number Theorem

Theorem 1 ([3]). *Let $\pi(x)$ denote the number of primes less than or equal to $x \geq 1$. The prime number theorem states that*

$$\lim_{x\to\infty} \frac{\pi(x)\ln x}{x} = 1$$

That is, $\pi(x) \sim \dfrac{x}{\ln x}$. This implies that n-th prime p_n is asymptotically given by $p_n \sim n \ln n$.

Moreover, we may frequently use these exact bounds on $\pi(x)$.

Lemma 2 ([3]). *For any $n \geq 2$,*

$$\frac{1}{6}\frac{n}{\ln n} \leq \pi(n) \leq 6\frac{n}{\ln n}$$

and for any $n \geq 1$,

$$\frac{1}{6} n \ln n \leq p_n \leq 12(n \ln n + n \ln \frac{12}{e})$$

2 Identification Codes

Identification is a communication paradigm introduced by Ahlswede [1]. In identification schemes, in essence, the receiver wants to know whether a certain message has been send or not. This is in contrast to the Shannon's transmission paradigm where the receiver wants to know the content of the message.

More formally, the send and receiver both have the message set $\mathcal{M}$ and the receiver is interested in message $m \in \mathcal{M}$. Ofcourse, when the sender knows m, he can send a bit to indicate that he intends to send m or not. We may then assume that sender does not know m.

This problem can be trivially addressed by transmission codes, the receiver decodes the received code to $\hat{m}$ and then decides if $\hat{m} = m$. However, the Ahlswede's identification codes require exponentially shorter blocklength to identify the same number of messages. This improvement is achieved mainly by relaxing the condition that the decoding sets need be disjoint. By allowing the decoding sets to have slight overlap, Ahlswede [1] has shown that there exists coding schemes that can identify $2^{2^{nC}}$ messages where C is the Shannon capacity of the DMC channel.

There are two kinds of errors associated with an identification scheme. The first kind happens when the sender sends m but the receiver fails to identify it and hence *misses* the identification. The second kind happens when the sender send $m' \neq m$ and the receiver *falsely* identifies m instead.

Definition (Identification code): An identification code $(n, N, \lambda_1, \lambda_2)$ for a DMC channel $\mathcal{W}^n(\mathcal{X}^n|\mathcal{Y}^n)$ is a set $\{(Q(\cdot|i), \mathcal{D}_i)\}_{i\in[N]}$ where $Q(\cdot|i)$ is a distribution over $\mathcal{X}^n$ to that encodes i – for determinstic encoder $Q(x_i|i) = 1$ for some $x_i \in \mathcal{X}^n$, and $\mathcal{D}_i \subset \mathcal{Y}^n$ is the decoding set of i. The first and second kind errors are bounded by λ_1 and λ_2, respectively.

$$P_{e,1}(i) = \sum_{x^n\in\mathcal{X}^n} Q(x^n|i)\mathcal{W}^n(\mathcal{D}_i^c|x^n) \leq \lambda_1$$
$$P_{e,2}(i,j) = \sum_{x^n\in\mathcal{X}^n} Q(x^n|j)\mathcal{W}^n(\mathcal{D}_i|x^n) \leq \lambda_2$$

For all $1 \leq i, j \leq N$ and $j \neq i$.

3 3-step Algorithm

The 3-step algorithm as described in [2] defines the following parameters.

- Let $\mathcal{M} = \{1, 2, \dots, M\}$ be the message set and $\alpha > 1$. Let $K = \lceil (\lg M)^{\alpha} \rceil$.
- Let π_i denote the i_{th} prime. Let $\mathcal{M}' = \{1, 2, 3, \dots, \pi_K\}$ and $K' = \lceil (\lg \pi_K)^{\alpha} \rceil$.
- Let us denote the set $\{1, 2, \dots, \pi_l\}$ by $\mathbb{Z}_l^+$. Define the function $\phi_l : \mathbb{N} \to \mathbb{Z}_l^+$ as follows.

$$\phi_l(n) = [n \mod \pi_l] + 1 \tag{17}$$

  Where $[n \mod \pi_l]$ is equal to the remainder of the division of n by π_l.

A round of communication in this scheme is as follows.

1. The sender chooses a key $k \leftarrow \mathcal{K} = \{1, 2, \dots, K\}$ uniformly and transmits it.
2. The sender chooses another key $l \leftarrow \mathcal{K}' = \{1, 2, \dots, K'\}$ uniformly and transmits it.
3. Given a message $m \in \mathcal{M}$, the sender transmits $\phi_l(\phi_k(m))$. Assuming that receiver wishes to identify $\hat{m} \in \mathcal{M}$, he calculate $\phi_l(\phi_k(\hat{m}))$ and compares it with $\phi_l(\phi_k(m))$. He identifies the message as $\hat{m}$ whenever $\phi_l(\phi_k(m))$ $= \phi_l(\phi_k(\hat{m}))$.

Example 1. Suppose $M = 100$ and $\alpha = 1.5$. Then, $K = 18$, $\pi_K = 61$, and $K' = 15$. Assume that $k = 12$ and $l = 7$ are chosen, with $\pi_k = 31$ and $\pi_l = 17$. Given $m = 71$, the sender sends the following code.

$$\underbrace{1100}_{k=12}\underbrace{111}_{l=7}\underbrace{01011}_{\phi_l(\phi_k(71))=11}$$

We immediately find this scheme to be problematic when the sender sends the code all at once. There needs to be some separator that allows the receiver to determine where each part of the code starts and ends. However, let us continue with the example. If $\hat{m}_1 = 71$, then clearly the receiver correctly identifies the message. If $\hat{m}_2 = 32$

$$\phi_l(\phi_k(\hat{m}_2)) = 16 = (10000)$$

and the receiver correctly does not identify the message. However, when $\hat{m}_3 = 9$, $\phi_l(\phi_k(\hat{m}_3)) = 11 = (01011)$ which means the receiver falsely identifies $m = 71$ as $\hat{m} = 9$.

Theorem 3 ([2]). *The 3-step algorithm produces a $(n = n(M, \alpha), M, \lambda_1 = 0, \lambda_2)$ identification code – per Definition 1– such that $\lambda_2 \to 0$ as $M \to \infty$. Moreover, this coding scheme is optimal.*

$$\lim_{M \to \infty} \frac{\lg \lg M}{n(M)} = \frac{1}{\alpha} \tag{18}$$

Proof. It is obvious that $\lambda_1 = 0$, that is the receiver will not miss an identification. For the second kind error probability consider the following lemmas.

Lemma 4. *Any positive integers m has at most $\lfloor \lg m \rfloor$ unique prime factors.*

Proof. Suppose $p_1, \dots, p_k$ are all the prime factors of m. Then for some $\alpha_1, \dots, \alpha_k \geq 1$

$$m = \prod_{i=1}^{k} p_i^{\alpha_i} \geq \prod_{i=1}^{k} 2^{\alpha_i} \geq 2^k$$

As a result, $k \leq \lfloor \lg m \rfloor$ as required. ■

Lemma 5. *For any $m, m' \in \mathcal{M} = \{1, 2, \dots, M\}$ such that $m \neq \hat{m}$*

$$\left| \left\{ k \in \{1, 2, \dots, K\} \,\middle|\, \phi_k(m) = \phi_k(\hat{m}) \right\} \right| \leq \lg M \tag{19}$$

Proof. The given set is the set of common prime factors of m and $\hat{m}$ that are less than or equal to π_K. The inequality immediatly follows from the fact that $m, \hat{m} \leq M$ and M has at most $\lg M$ prime factors. ■

We can derive an upper bound for the second kind error.

$$
\begin{aligned}
P_{e,2}(m,\hat{m}) &= \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid m \neq \hat{m}] && (20)\\
&= \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) = \phi_k(\hat{m}), m \neq \hat{m}] \Pr[\phi_k(m) = \phi_k(\hat{m}) \mid m \neq \hat{m}] && (21)\\
&\qquad \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) \neq \phi_k(\hat{m}), m \neq \hat{m}] \Pr[\phi_k(m) \neq \phi_k(\hat{m}) \mid m \neq \hat{m}] && (22)\\
&\leq \Pr[\phi_k(m) = \phi_k(\hat{m}) \mid m \neq \hat{m}] + \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) \neq \phi_k(\hat{m})] && (23)\\
&\leq \frac{\lg M}{K} + \frac{\lg M'}{K'} && (24)\\
&= \frac{\lg M}{\lceil (\lg M)^\alpha \rceil} + \frac{\lg \pi_K}{\lceil (\lg \pi_K)^\alpha \rceil} && (25)\\
&= \frac{1}{(\lg M)^{\alpha-1}} + \frac{1}{(\lg \pi_K)^{\alpha-1}} && (26)
\end{aligned}
$$

By the prime number theorem, Theorem 1, $\pi_K \sim K \ln K$. As a result, $\lambda_2 \to 0$ as $M \to \infty$.

$$(\lg \pi_K)^{\alpha-1} \sim (\lg K + \lg\lg K - \lg\lg e)^{\alpha-1} \approx (\alpha \lg\lg M + \lg\lg\lg M + \lg \alpha - \lg\lg e)^{\alpha-1}$$

Finally, the blocklength is calculated by $n = \lceil \lg K \rceil + \lceil \lg K' \rceil + \lceil \lg \pi_{K'} \rceil$. By applying the prime number theorem 2

$$
\begin{aligned}
n &= \lceil \lg K \rceil + \lceil \lg K' \rceil + \lceil \lg \pi_{K'} \rceil\\
&= \lceil \lg \lceil (\lg M)^\alpha \rceil \rceil + \lceil \lg \lceil (\lg \pi_K)^\alpha \rceil \rceil + \lceil \lg \pi_{K'} \rceil\\
&\approx \alpha \lg\lg M + \alpha \lg\lg \pi_K + \lg \pi_{K'}\\
&\approx \alpha \lg\lg M + (1 + o(1)) \lg\lg\lg M + (\alpha + o(1)) \lg\lg \pi_K\\
&\approx \alpha (1 + o(1)) \lg\lg M
\end{aligned}
$$

which was what was wanted. ■

This scheme requires sender to have access to a prime generation algorithm that given K computes the first K primes. And it requires the receiver to have access to another prime generation algorithm that given an index k output π_k, the k_{th} prime. To the best of our knowledge both these algorithm are exponential (I think the best algorithm are subexponential in fact)[needs referencing]. To alleviate these inefficiencies we propose the following modifications.

3.1 Modified 3-step algorithm

Consider the following parameters.

- Let $\mathcal{M} = \{1, 2, \ldots, M\}$ be the message set and let $K = \lceil (\lg M)^\alpha \rceil$, $K' = \lceil (\lg K)^\alpha \rceil$, for some constant $\alpha > 1$.
- Let us denote the set $\{1, 2, \ldots, l\}$ by $\mathbb{Z}_l^+$. Define the function $\phi_l : \mathbb{N} \to \mathbb{Z}_l^+$ as follows.

$$\phi_l(n) = [n \mod l] + 1 \tag{27}$$

Where $[n \mod l]$ is equal to the remainder of the division of n by l.

A round of communication in this scheme is as follows.

1. The sender chooses a probabilistic prime k from the set $\mathcal{K} = \{1, 2, \ldots, K\}$ by a prime number generator and transmits it.
2. The sender chooses another probabilistic prime l from the set $\mathcal{K}' = \{1, 2, \ldots, K'\}$ by the same prime number generator and transmits it.
3. Given a message $m \in \mathcal{M}$, the sender transmits $\phi_l(\phi_k(m))$. Assuming that receiver wishes to identify $\hat{m} \in \mathcal{M}$, he calculates $\phi_l(\phi_k(\hat{m}))$ and compares it with $\phi_l(\phi_k(m))$. He identifies the message as $\hat{m}$ whenever $\phi_l(\phi_k(m))$ $= \phi_l(\phi_k(\hat{m}))$.

Theorem 6. *This coding scheme is optimal.*

$$\lim_{M\to\infty} \frac{\lg\lg M}{n(M)} = \frac{1}{\alpha} \tag{28}$$

Proof. The blocklength is calculated by $n = \lceil \lg |\mathcal{K}| \rceil + 2\lceil \lg |\mathcal{K}'| \rceil$.

$$\begin{aligned} n &= \lceil \lg |\mathcal{K}| \rceil + 2\lceil \lg |\mathcal{K}'| \rceil && (29)\\ &= \lceil \lg K \rceil + 2\lceil \lg K' \rceil && (30)\\ &\approx \alpha \lg\lg M + 2\alpha \lg\lg K && (31)\\ &\approx \alpha \lg\lg M + 2\alpha \lg\lg\lg M + 2\alpha \lg \alpha && (32)\\ &\approx \alpha(1+o(1)) \lg\lg M && (33) \end{aligned}$$

which was what was wanted. ■

The error analysis this code depends on the prime number generator. We will be using a simple prime number generator based on the Miller-Rabin primality test.

Theorem 7. *The modified 3-step algorithm produces a $(n = n(M, \alpha), M, \lambda_1 = 0, \lambda_2)$ identification code – per Definition 1– such that $\lambda_2 \to 0$ as $M \to \infty$. Moreover, this coding scheme is optimal.*

$$\lim_{M\to\infty} \frac{\lg\lg M}{n(M)} = \frac{1}{\alpha} \tag{34}$$

We employ the same techniques used by Ahlswede by first calculating the following probability.

Lemma 8. *Suppose $m, \hat{m} \in \mathcal{M} = \{1, 2, \ldots, M\}$ and $p \leftarrow \mathcal{A}(K, \epsilon)$ is a probabilistic prime generated by randomized algorithm $\mathcal{A}$ such that all primes less than or equal to K are generated equiprobably and*

$$\Pr[p \notin \mathcal{P}] \leq \epsilon$$

Then,

$$\Pr[\phi_p(m) = \phi_p(\hat{m}) \mid m \neq \hat{m}] \leq \frac{\lg M}{\pi(K) \lg\lg M} + \epsilon$$

Proof. Firstly, note that the least number that has k unique prime factors is $U_k = 2 \cdot 3 \cdots \pi_k = \prod_{i=1}^{k} \pi_i$. Therefore, if $M < U_k$, then M has at most $k-1$ different prime factors. Moreoever,

$$U_k \geq 1 \cdot 2 \cdot 3 \cdots k = k! \tag{35}$$

Suppose, k is such that $M < k!$, then $M < U_k$ and thus for all $m \leq M$, m has at most $k-1$ different prime factors. In other words,

$$\frac{|\{\pi \leq K \mid \phi_\pi(m) = \phi_\pi(\hat{m})\}|}{|\{\pi \leq K\}|} \leq \frac{k-1}{\pi(K)} \tag{36}$$

To simplify our notation, let $!M$ denote the inverse of factorial i.e. the largest integer k such that $k! \leq M$. Then,

$$\frac{|\{\pi \leq K \mid \phi_\pi(m) = \phi_\pi(\hat{m})\}|}{|\{\pi \leq K\}|} \leq \frac{!M}{\pi(K)} \tag{37}$$

By Stirling approximation

$$!M = \mathcal{O}\left(\frac{\lg M}{\lg\lg M}\right) \tag{38}$$

and by prime number theorem

$$\pi(K) = \frac{K}{\lg K}$$

As a result,

$$\begin{aligned} \Pr[\phi_p(m) = \phi_p(\hat{m}) \mid m \neq \hat{m}] &= \Pr[\phi_p(m) = \phi_p(\hat{m}) \mid p \in \mathcal{P}, m \neq \hat{m}] \Pr[p \in \mathcal{P}] && (39)\\ &\quad + \Pr[\phi_p(m) = \phi_p(\hat{m}) \mid p \notin \mathcal{P}, m \neq \hat{m}] \Pr[p \notin \mathcal{P}] && (40)\\ &\leq \Pr[\phi_p(m) = \phi_p(\hat{m}) \mid p \in \mathcal{P}, m \neq \hat{m}] + \epsilon && (41)\\ &\leq \frac{!M}{\pi(K)} + \epsilon && (42)\\ &\leq \frac{\lg M}{\pi(K) \lg\lg M} + \epsilon && (43)\\ &\approx \frac{\lg M \lg K}{K \lg\lg M} + \epsilon && (44) \end{aligned}$$

■

The proof of Theorem 7 works as follows.

Proof. Let $m, \hat{m} \in \mathcal{M} = \{1, 2, \ldots, M\}$ be given such that $m \neq \hat{m}$. Let $\mathcal{A}$ be the randomized prime generator as described above. Suppose the probable primes $k \leftarrow \mathcal{A}(K, \epsilon)$ and $l \leftarrow \mathcal{A}(K', \epsilon)$ are generated with $K = \lceil (\lg M)^{\alpha} \rceil$ and $K' = \lceil (\lg K)^{\alpha} \rceil$. The probability of the second kind error is given as follow

$$
\begin{aligned}
P_{e,2}(m, \hat{m}) &= \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid m \neq \hat{m}] && (45)\\
&= \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) = \phi_k(\hat{m}), m \neq \hat{m}] \\
&\quad\ \Pr[\phi_k(m) = \phi_k(\hat{m}) \mid m \neq \hat{m}] \\
&\quad + \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) \neq \phi_k(\hat{m}), m \neq \hat{m}] && (46)\\
&\quad\ \Pr[\phi_k(m) \neq \phi_k(\hat{m}) \mid m \neq \hat{m}] \\
&\leq \Pr[\phi_k(m) = \phi_k(\hat{m}) \mid m \neq \hat{m}] + \Pr[\phi_l(\phi_k(m)) = \phi_l(\phi_k(\hat{m})) \mid \phi_k(m) \neq \phi_k(\hat{m})] && (47)\\
&\leq \frac{\lg M}{\pi(K) \lg\lg M} + \frac{\lg K}{\pi(K') \lg\lg K} + 2\epsilon && \text{(Lemma 8)} \quad (48)\\
&\approx \frac{\lg(M)\lg(K)}{K \lg\lg M} + \frac{\lg(K)\lg(K')}{K' \lg\lg K} + 2\epsilon && \text{(PNT)} \quad (49)\\
&\approx \frac{\alpha \lg\lg M}{(\lg M)^{\alpha-1} \lg\lg M} + \frac{\alpha \lg\lg K}{(\lg K)^{\alpha-1} \lg\lg K} + 2\epsilon && (50)\\
&= \frac{\alpha}{(\lg M)^{\alpha-1}} + \frac{\alpha}{(\lg K)^{\alpha-1}} + 2\epsilon && (51)
\end{aligned}
$$

Since ϵ can be chosen as small as possible, then $\lambda_2 \to 0$ as $M \to \infty$. ■

3.2 Performance

As oppose to the original 3-step algorithm of Ahlswede, in our revision most of the work is done by the transmitter. Specifically, the transmitter runs the prime number generator $\mathcal{A}$ twice. Suppose the simple Algorithm 2 is used with Miller-Rabin primality test. By the analysis given in the Section 1.1, this algorithm in average runs in $\mathcal{O}\Big((\lg n)^4\Big)$ where n is its input. Therefore, on input K, the algorithm runs in

$$\mathcal{O}\Big((\lg K)^4\Big) = \mathcal{O}\Big((\alpha \lg\lg M)^4\Big) = \mathcal{O}\big(n^4\big)$$

and on input K', it runs in

$$\mathcal{O}\Big((\lg K')^4\Big) = \mathcal{O}\Big((\alpha \lg\lg K)^4\Big) = \mathcal{O}\Big((\lg n)^4\Big)$$

All together, we have significantly improved time complexity of the scheme without affecting its error probability or blocklength.

3.3 Simulation

We have implemented a code that mainly runs the Algorithm 2 to produce two probable prime numbers. Then it calculates the blocklength, error probability, and the running time of the simulation. On $\alpha = 1.01$ we get the following curves.

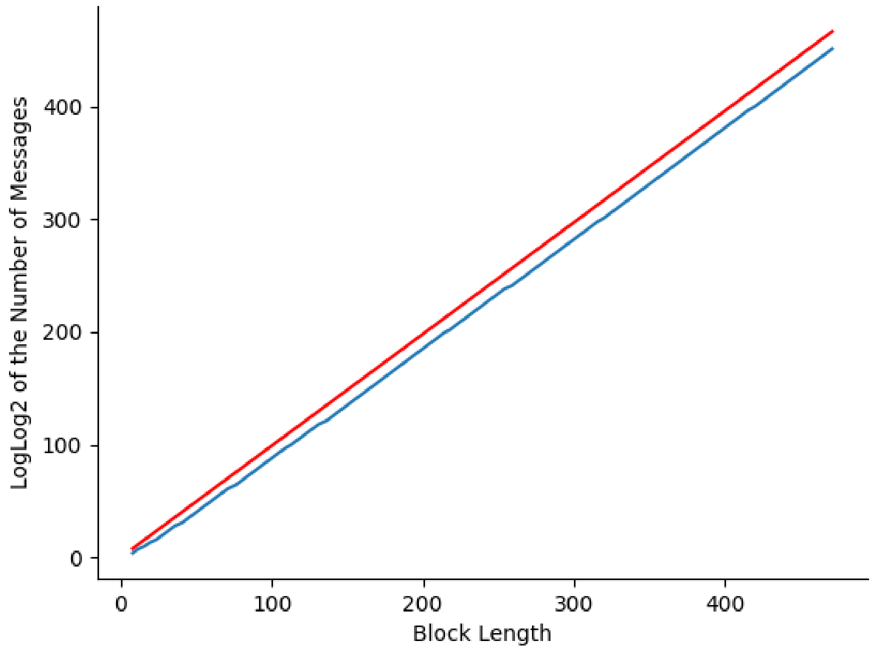

Figure 1: The lg lg of the number of message vs block length curve

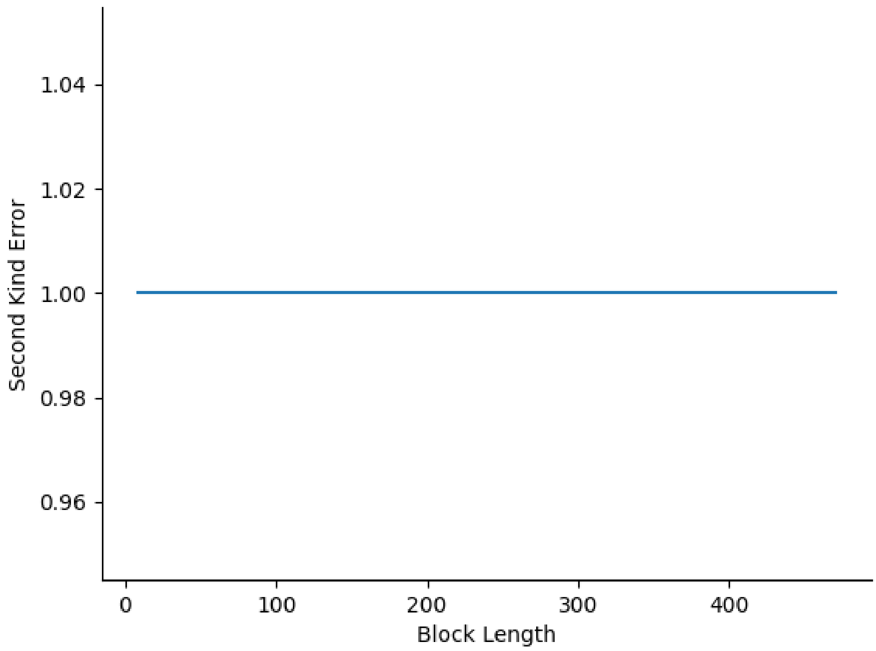

Figure 2: The average second kind error vs block length curve

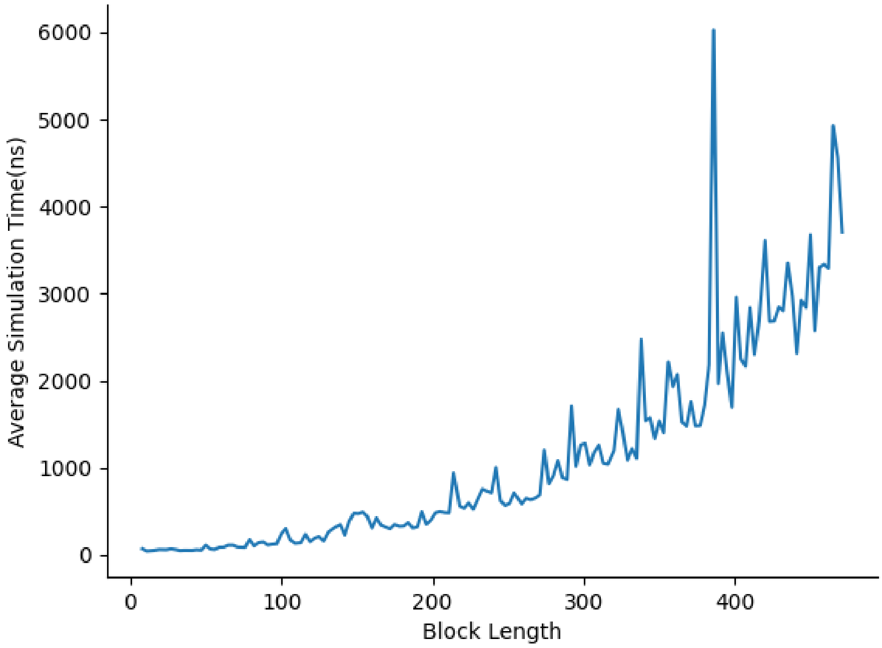

Figure 3: The average running time vs block length curve

References

[1] R. Ahlswede and G. Dueck. Identification via channels. *IEEE Transactions on Information Theory*, 35(1):15–29, 1989.

[2] R. Ahlswede and B. Verboven. On identification via multiway channels with feedback. *IEEE Transactions on Information Theory*, 37(6):1519–1526, 1991.

[3] Tom M. Apostol. *Introduction to Analytic Number Theory*. Springer New York, 1976.

[4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms, Third Edition*. The MIT Press, 3rd edition, 2009.