# Post Quantum Cryptography Performance Results

CSCE 482 Capstone

Texas A&M University

Authors:

Josef Munduchirakal, Aidan Heffron, Grant Shields, Shifan Hirani, and Blake Lun

## Table of Contents

# 1. Overview

Our project consisted of creating a client and server application that utilizes Post-Quantum Cryptography to authenticate and encrypt a connection between two different hosts.

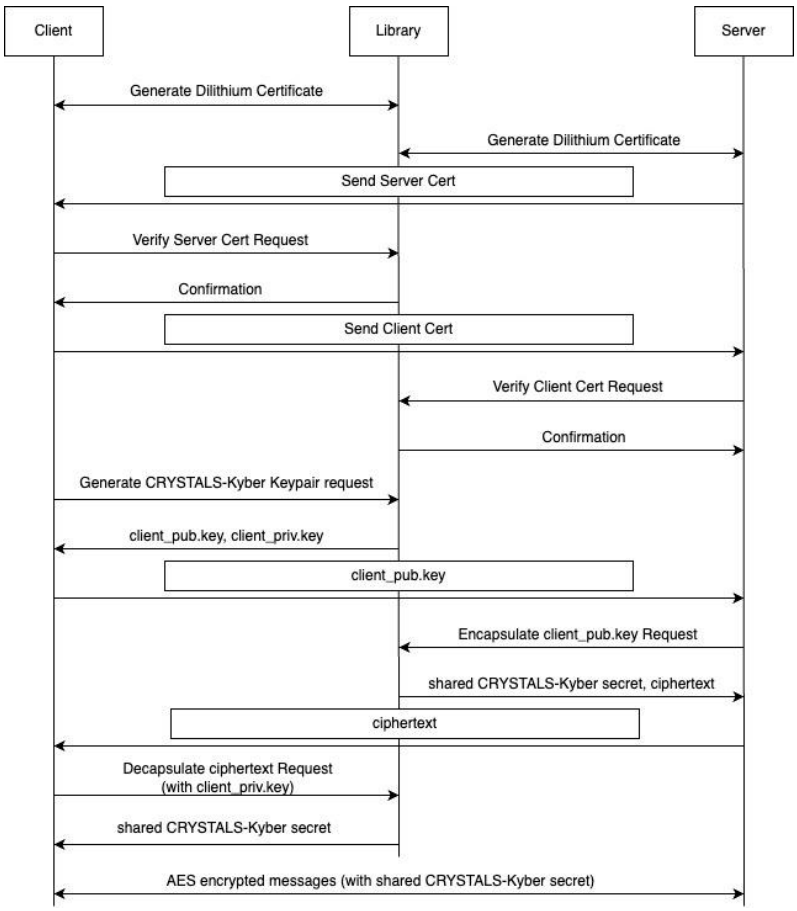


**Figure 1.1** Client/Server Sequence Diagram

**Figure 1.1** above shows how the client and server handle the security when a connection is established.

To sufficiently test our program, we used several network topologies, each aimed at a different real-world scenario. Within each topology, we performed five trials for different configurations of Singing Algorithms and KEM algorithms. We measured the time it took to complete different essential aspects of client authentication, key exchange, and encryption.

With all this data, we can directly compare how Post-Quantum Cryptography affects clients and servers.

The different signing algorithms we chose to test are as follows:

- Dilithium3
- P521_Dilithium5
- RSA

For each signing algorithm, we also used three different KEM algorithms:

- Kyber512
- Kyber768
- EC-DH

This set of different signing and KEM algorithms gives us a chance to test how the network is affected by post-quantum signing algorithms with post-quantum KEM algorithms, post-quantum signing algorithms with the commonly used classical EC-DH algorithm, and classical RSA singing with post-quantum KEM.

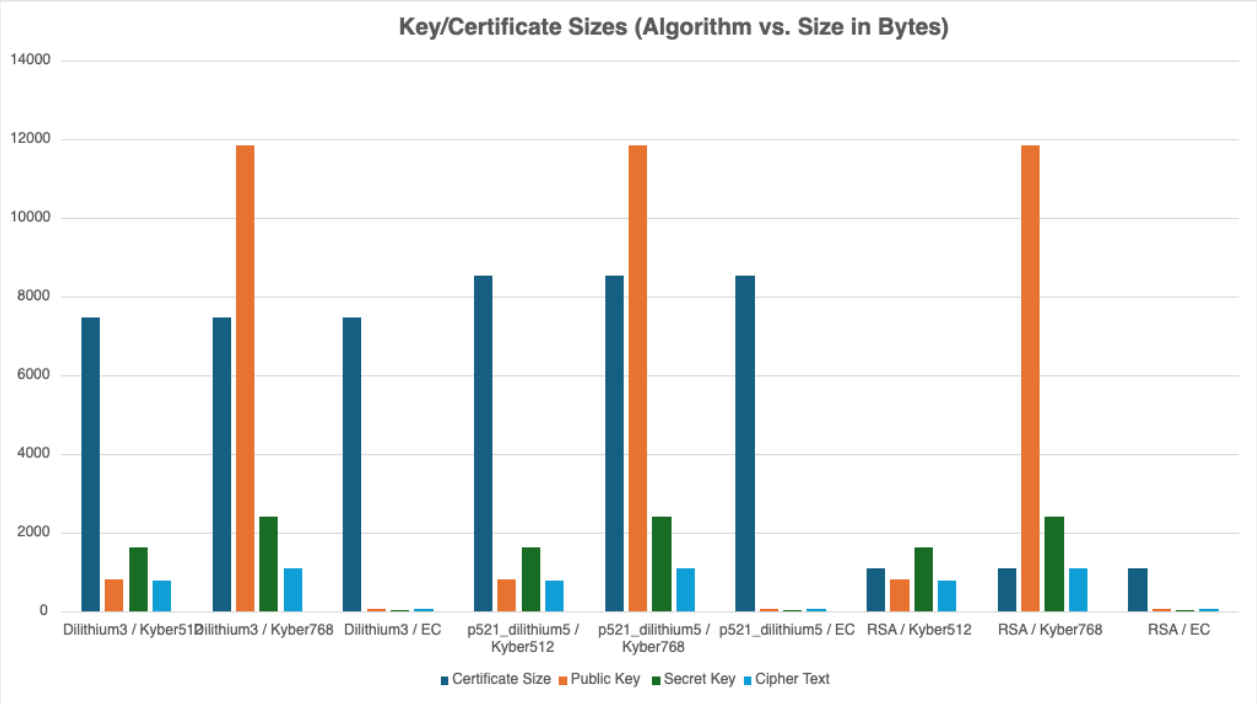


**Figure 1.2** Key/Certificate Sizes for different Permutations

Looking at **Figure 1.2** we can quickly see how much larger the post-quantum certificates and keys are than the classical. This information works to prove that we will likely run into increased delays when writing information about a certificate or key between applications when using a post-quantum algorithm.

We tested all permutations of the above algorithms on the following network topologies:

- New York to Los Angeles
  - Packet loss = 0.5%, latency = 110ms
- New York to Tokyo
  - Packet loss = 0.5%, latency = 190ms
- Low, Average, and High Bandwidth
  - Packet loss = 0.5%, latency = 10ms, bandwidth = 25Mbps, 200Mbps, 450Mbps
- Low and High Packet Loss
  - Packet loss = 1%, 10%, latency = 10ms
- Low Earth Orbit Satellite
  - Packet loss = 1%, latency = 25ms, bandwidth = 300Mbps
- Geosynchronous Satellite
  - Packet loss = 5%, latency = 600ms, bandwidth = 200Mbps

Looking ahead, we will be diving into each topology more closely and looking at exactly how the change affects the performance of each run. We will explain any inconsistencies that may have appeared and talk about how we would mitigate these if given more time and resources to test.

## 2. New York to Los Angeles

A New York host communicating with a Los Angeles host would likely experience a delay of about 110ms. This delay is brought out by many different variables such as non-continuous networking cables. Along with this, there is an expected packet loss of about 0.5% for a strong fiber optic connection.

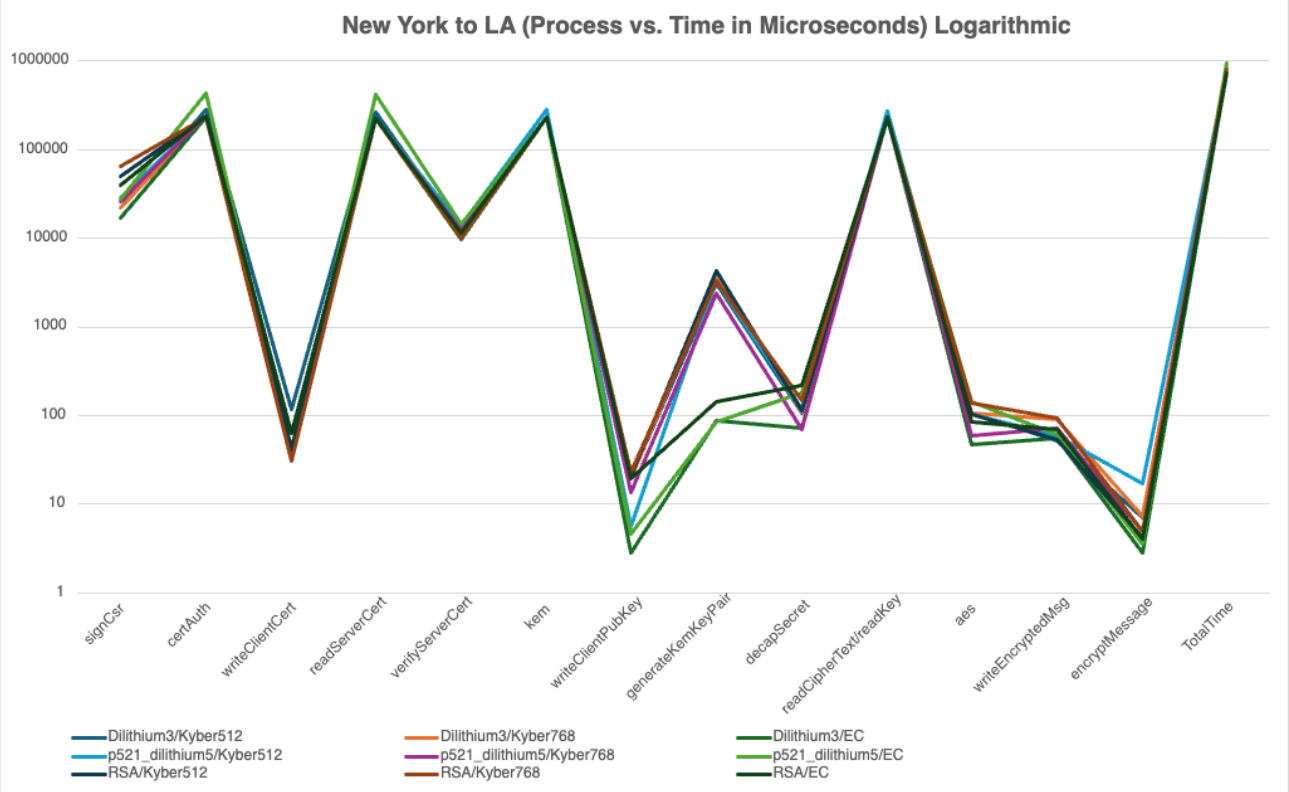


**Figure 2.1** New York to LA results

**Figure 2.1** is used to highlight how long in microseconds it takes to accomplish different parts of our program. We can see that overall, the different algorithms perform about the same in terms of total time to create the connection. The biggest difference comes in how long it takes to generate the different cryptographic elements (certificates and keys) and how long it takes to transfer these elements.

It seems that, as expected, classical algorithms take less time to write over public key and certificate information.

# 3. New York to Tokyo

A New York host communicating with a Tokyo host would likely experience a delay of about 190ms. Along with this, there is an expected packet loss of about 0.5% for a strong fiber optic connection.

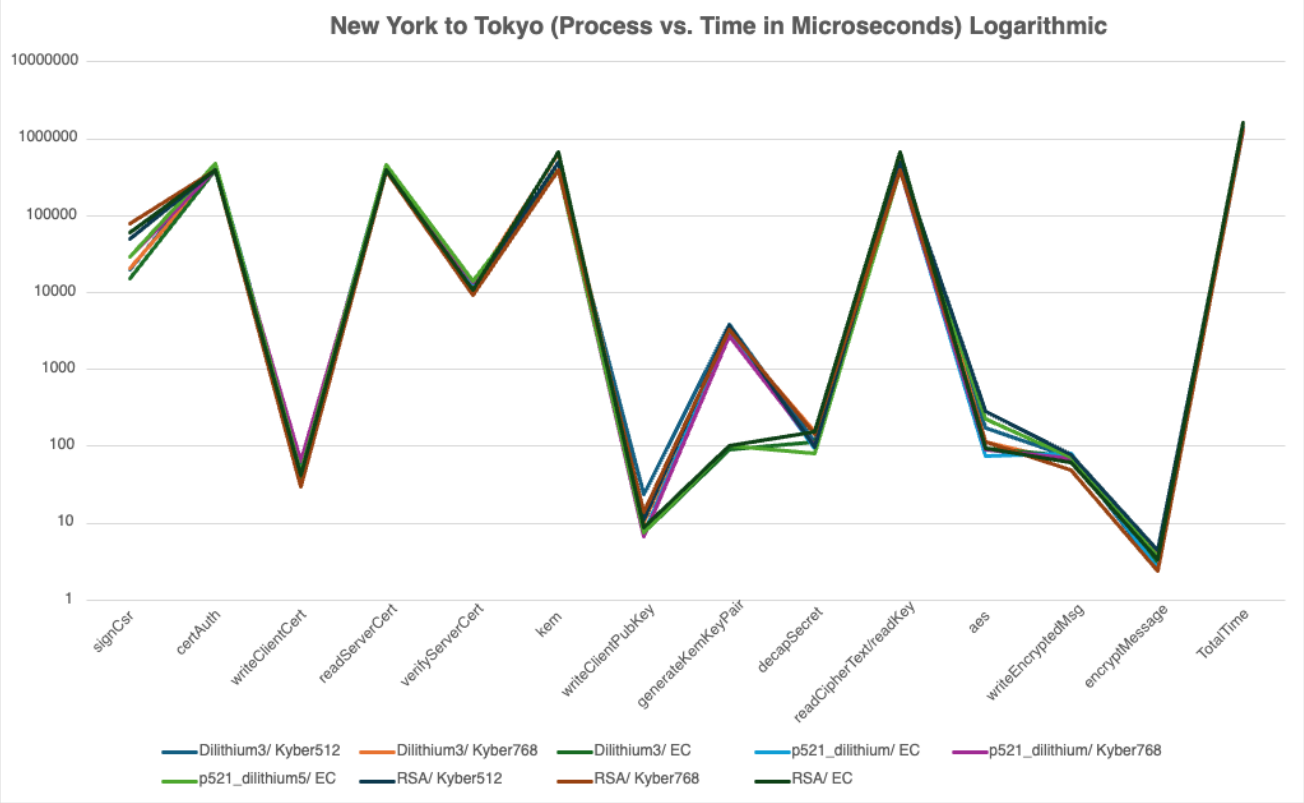


**Figure 3.1** New York to Tokyo Results

**Figure 3.1** Shows very similar results to the New York to LA example in the previous section. The different algorithms run at about the same speed but use their time in different areas. Classical algorithms tend to be quicker in writing their information from one host to another yet sometimes take longer to produce certificates and read in information. This difference in the size of the certificate and public keys does not seem to be that large of an issue yet.

The packets can carry a lot of information due to the unrestricted bandwidth of the connection and thus it takes about the same number of packets to bring over the data.

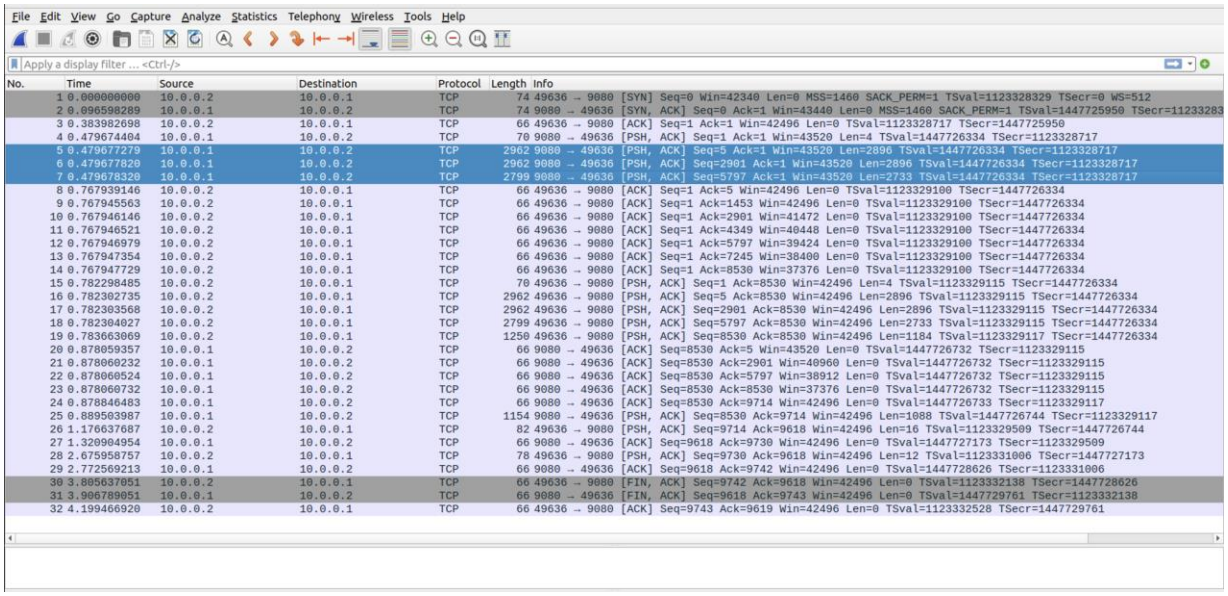


**Figure 3.2** P521_Dilithium5 / Kyber768 Wireshark

**Figure 3.2** Shows a wireshark capture that was caught during one of the trial runs. It shows how the entire 8,525-byte certificate can be transmitted from client to server in only 3 packets. Without needing to worry much about an unstable connection, increased packet loss, or anything like that, the speed at which this process happens is rather close to that of the RSA / EC output.

# 4. Low Bandwidth Connection

According to the Federal Communications Commission (FCC), a good internet connection should be able to download data at a rate of at least 25 Mbps. We have taken this as our low bandwidth connection. Along with this, we are assuming a 0.5% packet loss and a latency of 20ms.

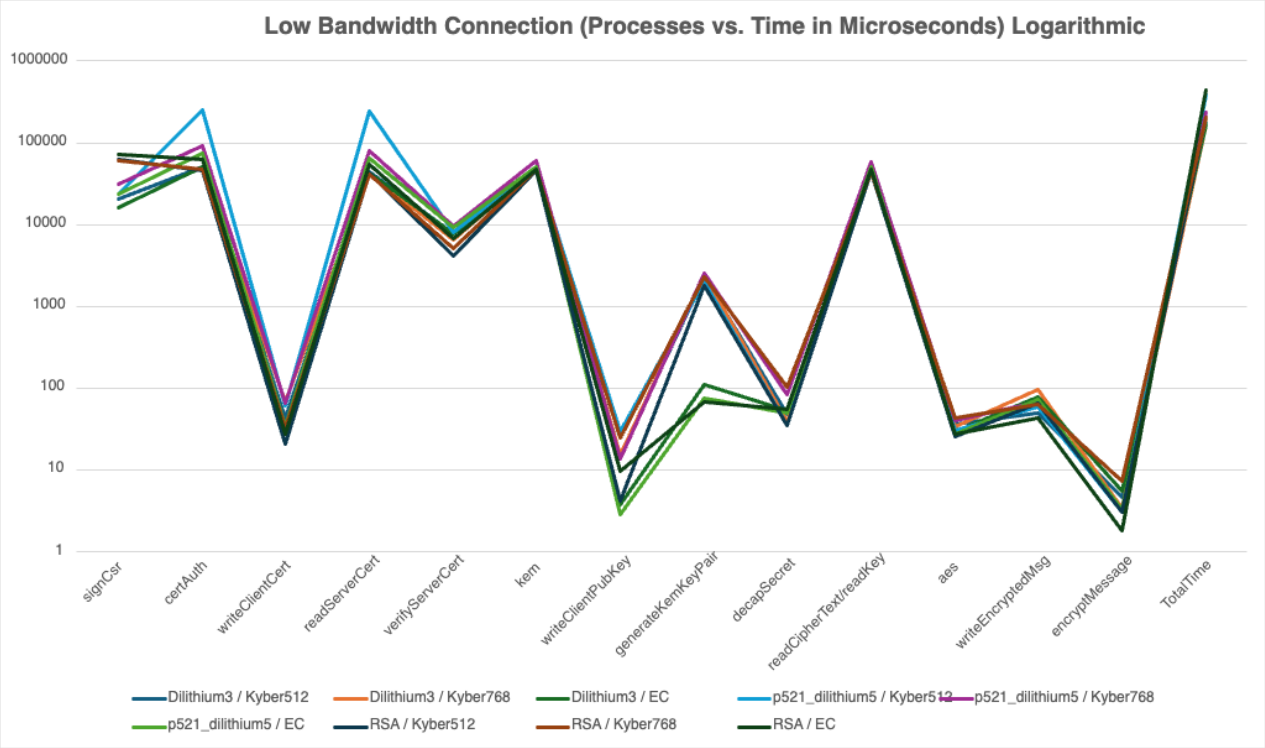


**Figure 4.1** Low Bandwidth results

Now, looking at **Figure 4.1** It is much easier to see that the classical algorithms begin to outperform their post-quantum counterparts. The difference is not too drastic, but the RSA signing algorithms tend to create and sign a CSR faster than Dilithium3 and p521_Dilithium5. The time it takes to write over the different cryptographic components is faster for RSA and EC than it is for the post-quantum algorithms.

The Low bandwidth causes more packets to be exchanged to send over the same amount of data. This makes it so the length of the post-quantum certificates and keys penalizes the performance more and more. It is expected that the lower the bandwidth, the more the classical algorithms will outperform the quantum-safe algorithms. This concept is explored more in sections 11 and 12.

# 5. Average Bandwidth Connection

The average download speed in the United States in 2022 was 204 Mbps, therefore we have taken 200Mbps as our average bandwidth. Along with this we are assuming a 0.5% packet loss rate and a latency of 20ms.

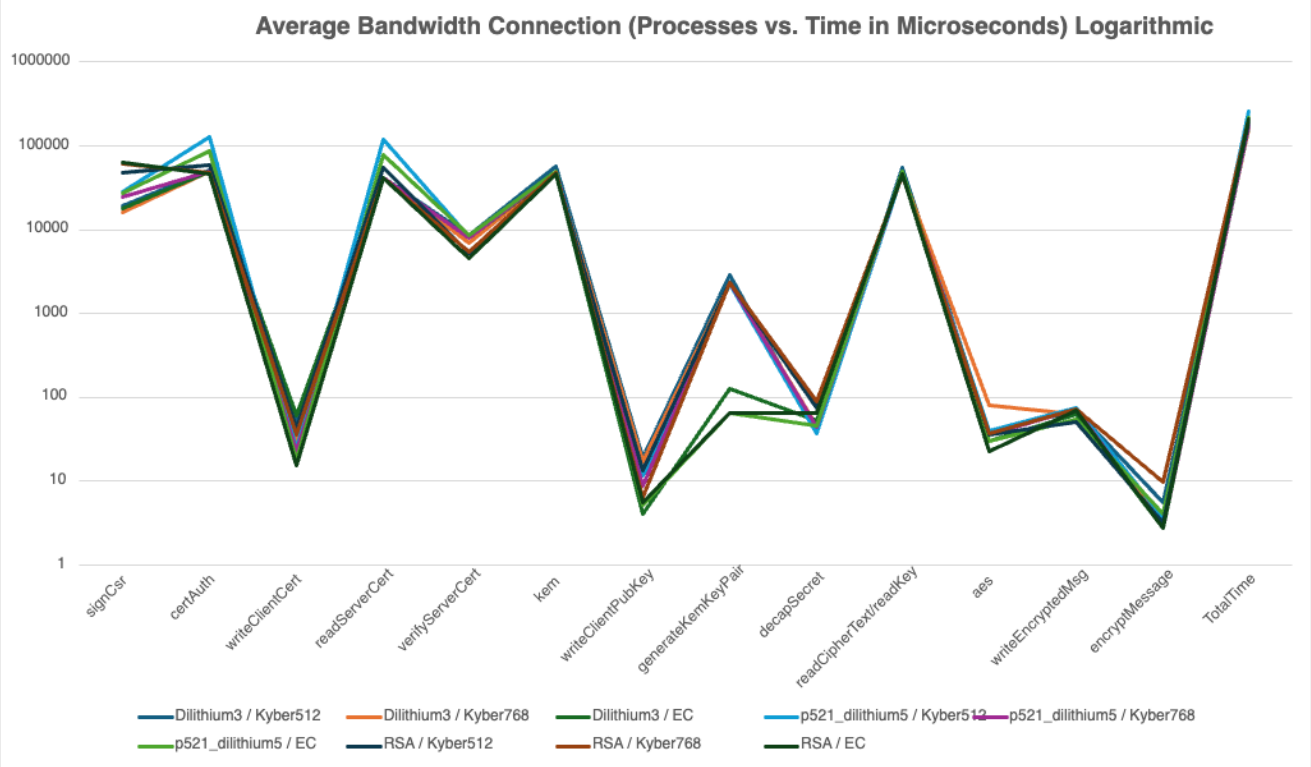


**Figure 5.1** Average Bandwidth Results

With the average bandwidth, we see the same trend that we have seen before. Classical algorithms outperform post-quantum algorithms due to the decreased time it takes to transmit data from one host to the other. The 200Mbps bandwidth still works to limit the size of the packets being sent from host to host meaning that the larger certificates and keys are taking longer to be read in.

It is interesting to note that with all the results we have seen so far, the RSA signing algorithm tends to take longer than some if not all the post-quantum algorithms. This means that although the RSA certificate is much smaller than the Dilithium ones, it takes a longer computation time to actually produce. Then it is transferred faster due to its smaller size.

# 6. High Bandwidth Connection

The fastest Wi-Fi offered by Texas A&M is 450Mpbs, so we have chosen to use this as our high bandwidth. Along with this is 20ms of latency and 0.5% packet loss.

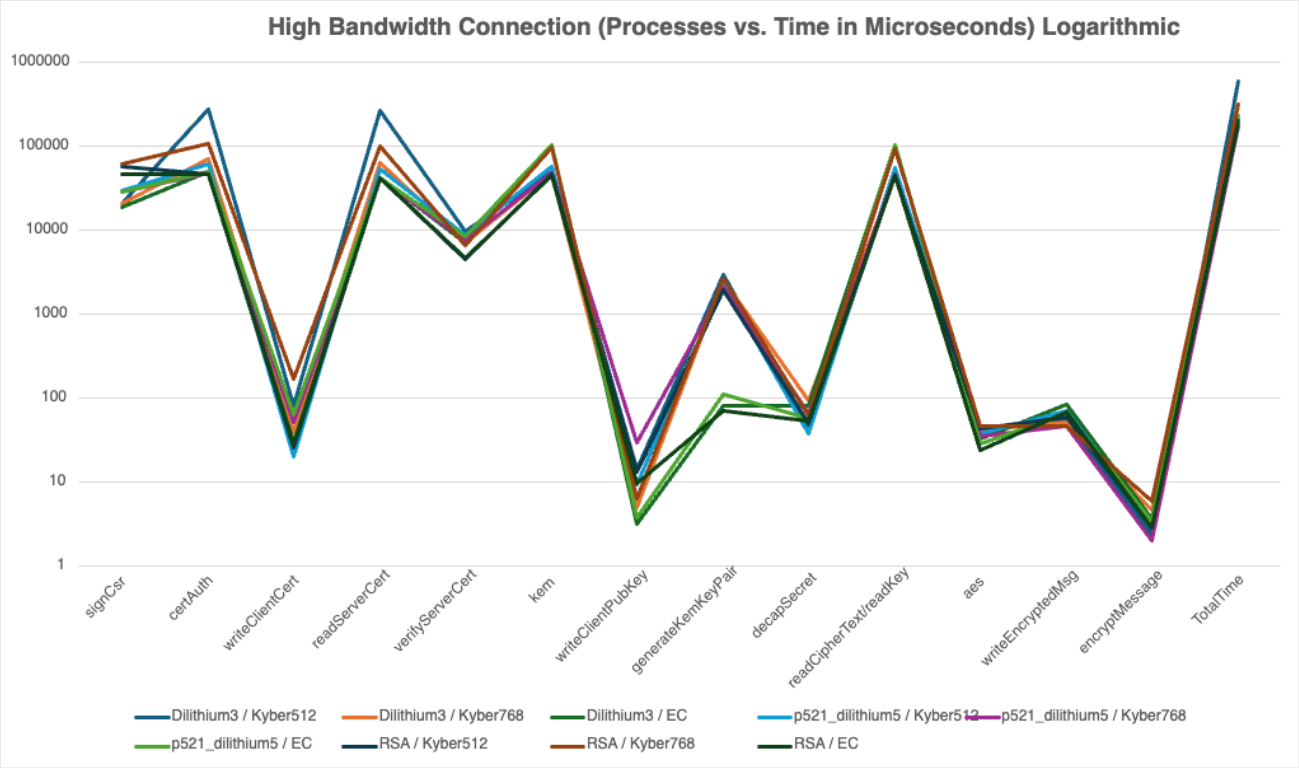


**Figure 6.1** High Bandwidth Results

**Figure 6.1** shows that the results of the high bandwidth connection are very similar to the results in the New York to LA and New York to Tokyo examples. The increased bandwidth allows packets to send over more information faster and thus the effect of having a large certificate or large KEM keys is not felt as much.

# 7. Low Earth Orbit Satellite

Satellites in low earth orbit tend to be anywhere from 100 to 1000 miles above the surface of the Earth. The expected latency is supposed to be around 25ms and the resulting packet loss around 1%. We are also assuming a bandwidth of 300 Mbps.

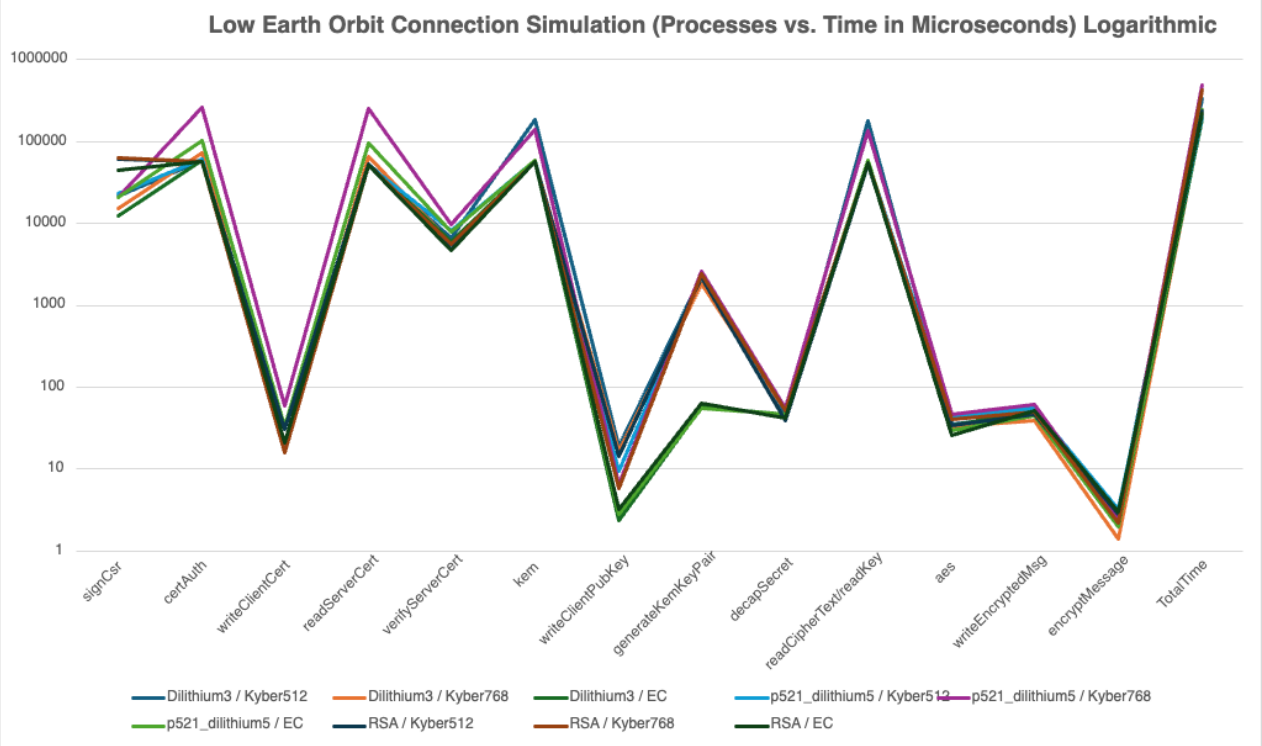


**Figure 7.1** Low Orbit Results

The effects of an increased packet loss (to 1%) and the overall latency start to play a bigger role in the results shown above. Generating and writing keys and certificates seems to be faster for the classical implementations. This effect is very evident in **Figure 7.1** at the "writeClientPubKey" field. We can see that all the EC KEM runs significantly outperformed the post-quantum runs.

Of course, generating RSA certificates is still a bit of a longer process due to the nature of the algorithm itself, however transmitting this certificate tends to take less time than the post-quantum implementations.

The logarithmic scaling of the figure belittles the overall effect, but the total time for the algorithms running with Kyber768 KEM tends to be about one and a half times the time of algorithms running EC-DH.

# 8. Low Packet Loss

Normally, in everyday connections we can see packet loss of about 0.5%, however there are cases where we can see about 1%, we are taking this as our low packet loss. Along with this is a latency of about 10ms

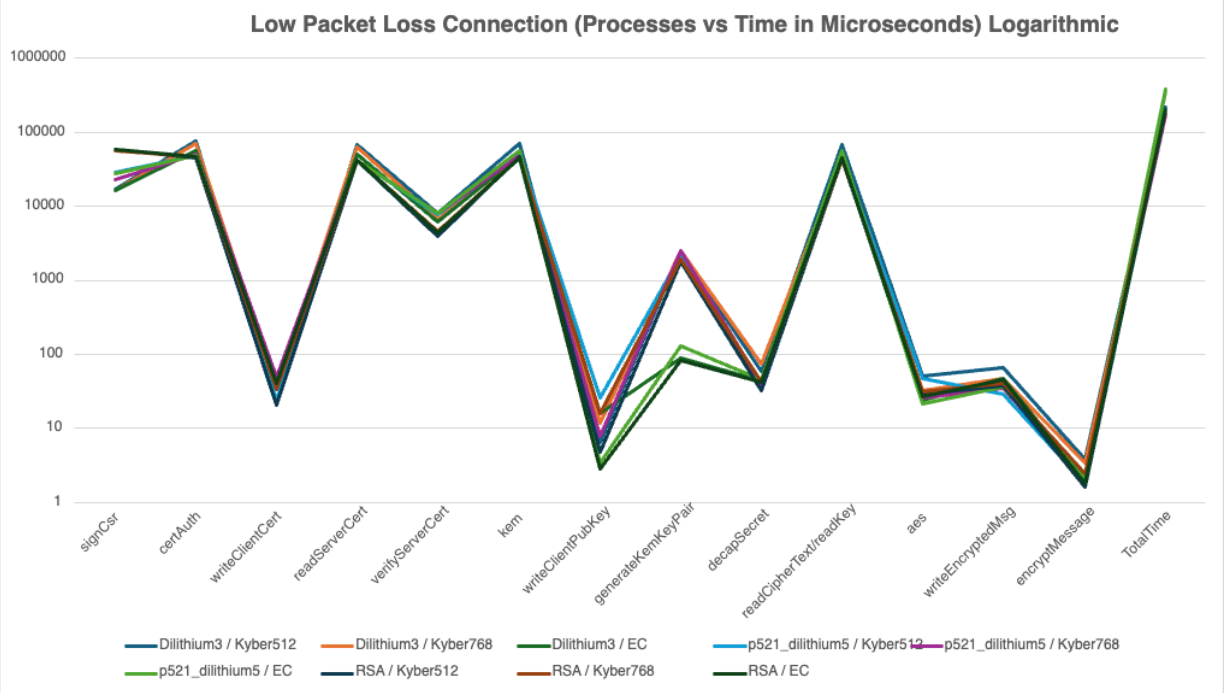


**Figure 8.1** Low Packet Loss Results

When running with low packet loss, it is no surprise that we see results rather similar to the results we have been seeing. The low percentage of packets getting lost means we do not need to retransmit the same information over and over and thus the effects of having larger certificates or keys is not felt as much.

# 9. High Packet Loss

In certain networks, large amounts of noise or other variables can cause a very congested network, in which packets are frequently lost. We take a packet loss of 10% for our congested network, along with 20ms of latency.

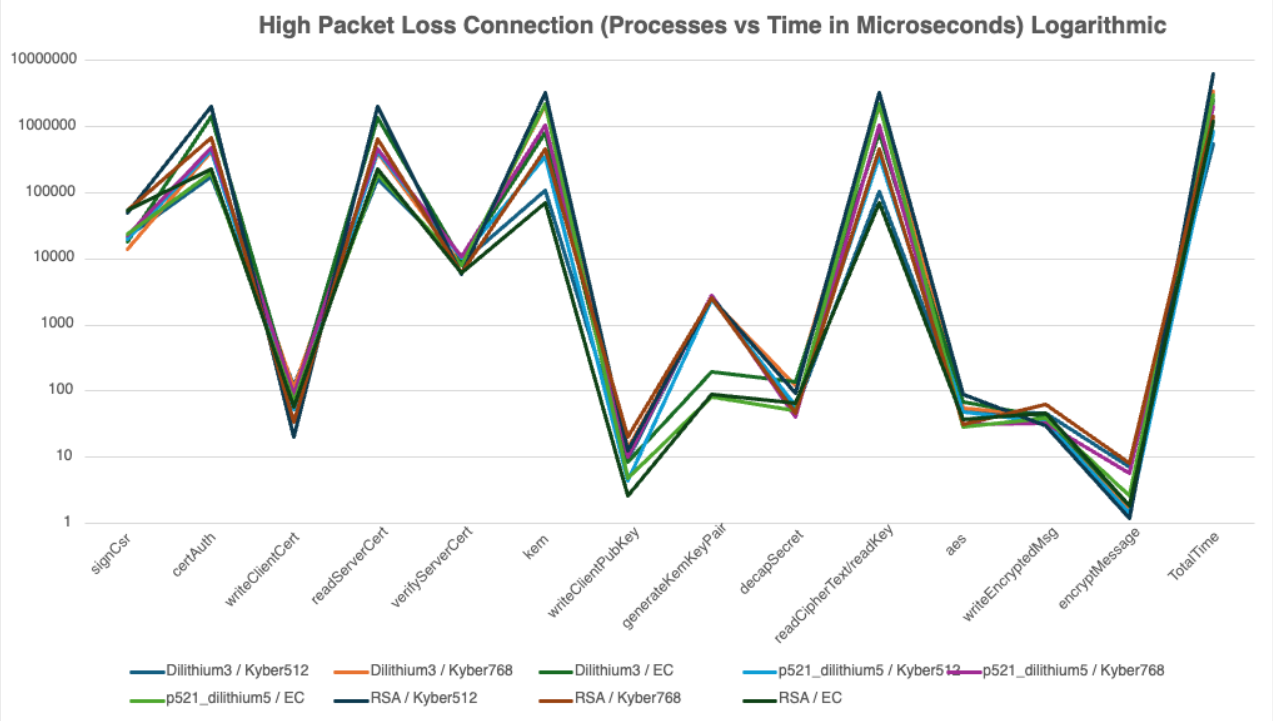


**Figure 9.1** High Packet Loss Results

The graph shown in **Figure 9.1** shows the devastating effects of congestion on a network. There is great variability in the time it takes the different algorithms to compute different elements. We see that there still exists the same trend as before wherein the classical components take less time to generate KEM keys, write keys, etc., but longer to generate RSA certificates. The time it takes to read different information is greatly varied because different packets are lost when transmitting information.

When we ran these tests, we took 5 trials and averaged them together. Looking at the above data, we believe that we would get more valuable information if we had time to run up to 30 trials and average them. This is because the drastic effects of up to 10% packet loss is highly variable and with only 5 data points per algorithm, it is hard to say if the algorithm is the reason it is taking the time it takes or if there were simply way more packets lost for one algorithm compared to another.

# 10. Geosynchronous Earth Orbit Satellite

Geosynchronous satellites orbit earth from a much larger distance than low earth orbit satellites. They are about 22,200 miles above the surface of the Earth. This large distance works to provide a latency of about 600ms and packet loss percentage of 5%. Along with this, we are assuming a bandwidth of 200Mbps.

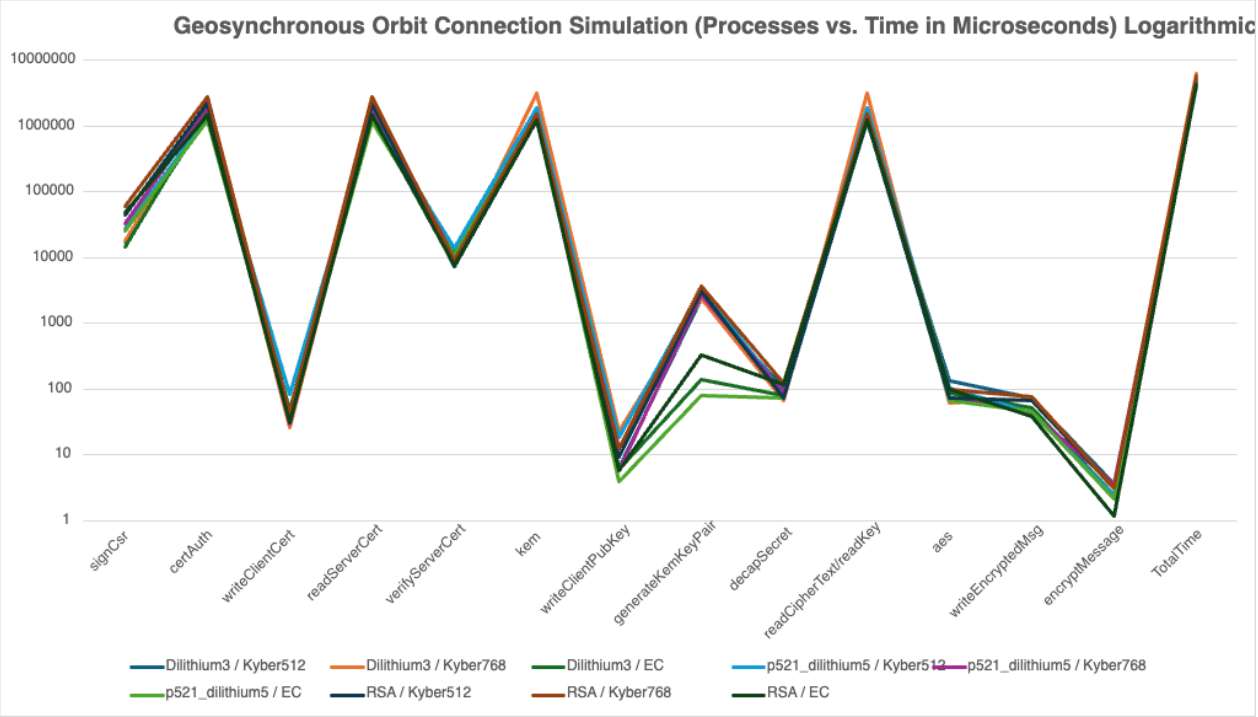


**Figure 10.1** Geosynchronous Orbit Results

The results shown in the figure above are harder to see on the Logarithmic scale, so we provide a linear scale in the figure below.

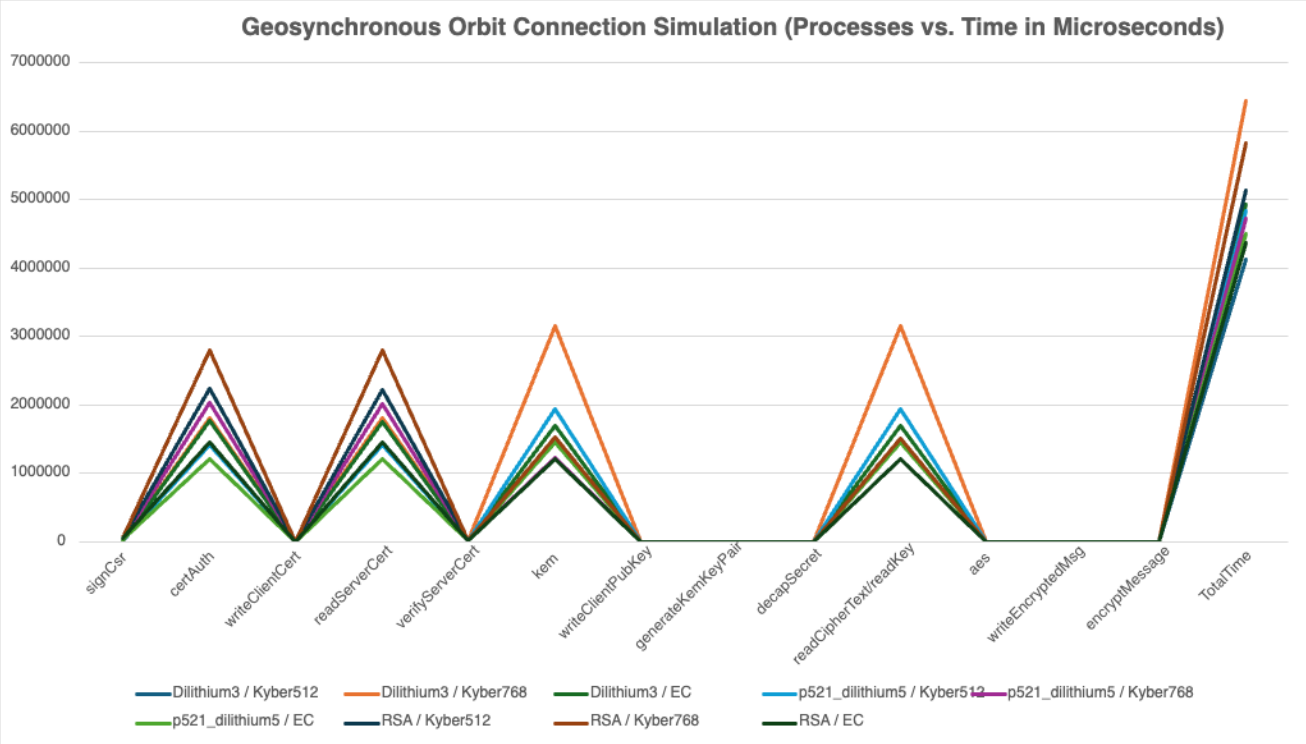


**Figure 10.2** Geosynchronous Orbit Linear Results

Looking at the above data it's easy to see how the classical implementations are sometimes performing twice as fast as the post-quantum implementations, mainly for the KEM process. Certificate authentication seems to take a while for the post-quantum implementations due to the time it takes to send over information from one host to the other.

# 11. Key Size vs Write Speed

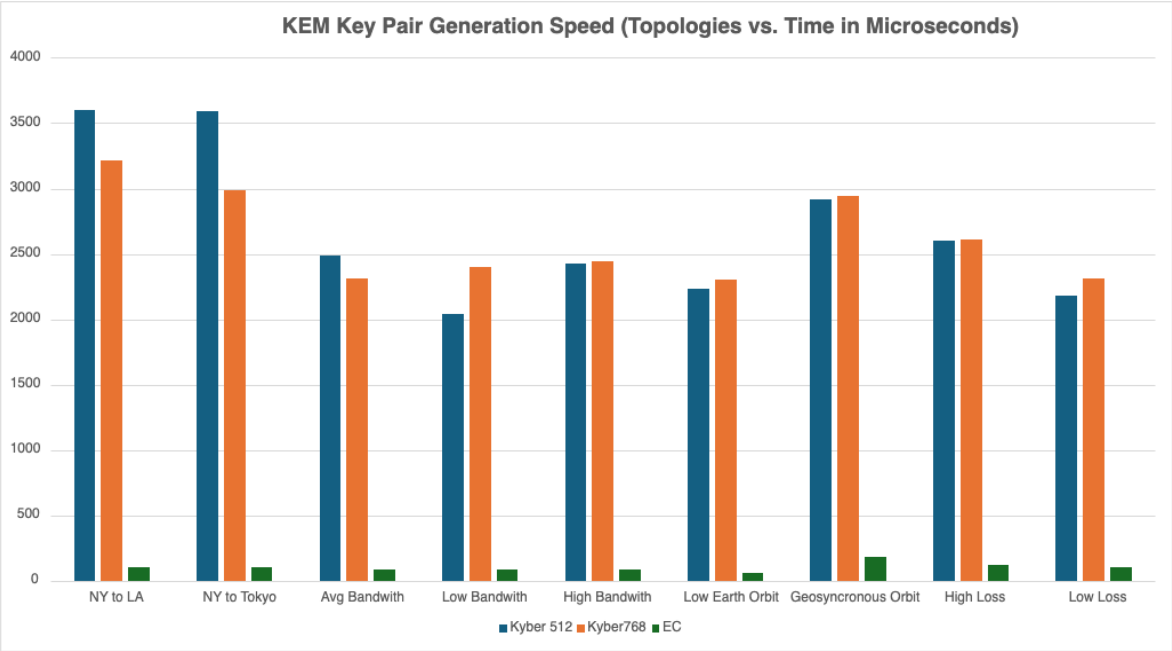


**Figure 11.1** Time to Generate KEM Key Pair

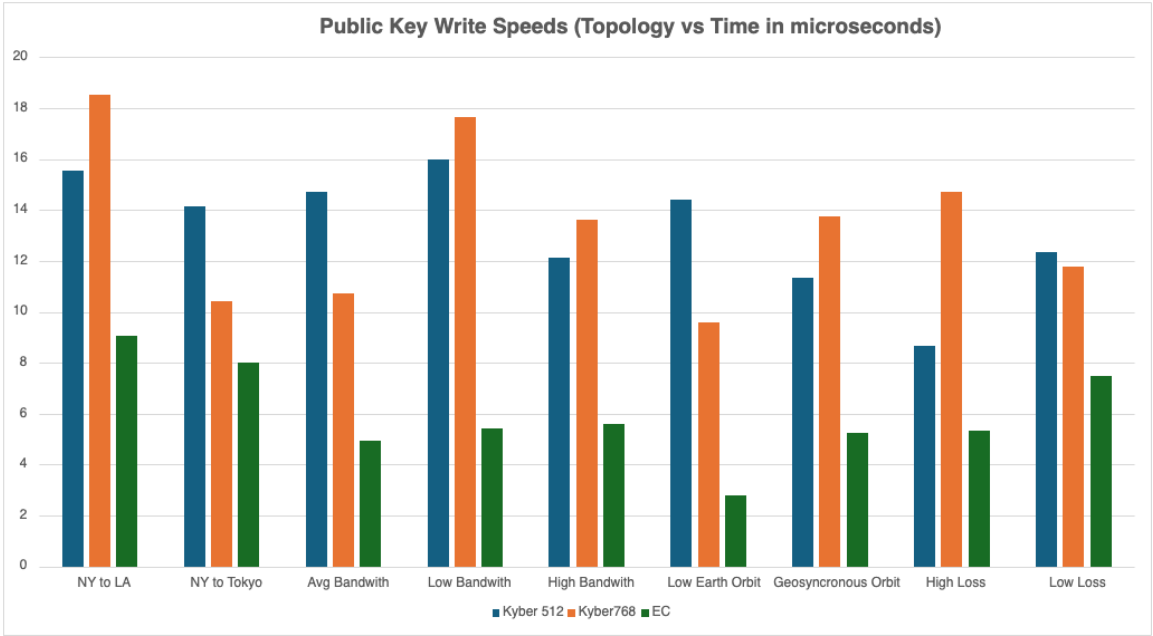


**Figure 11.2** Time to Send KEM Key

**Figure 11.1** and **Figure 11.2** go hand-in-hand in explaining why we see a lot of the trends presented above. The time it takes to generate an EC key pair is a fraction of the time it takes to generate a Kyber512 or Kyber768 key pair. Along with this, the generated key is

rather small and so it does not take much to send the key from one host to another. In situations like High Packet loss, this means that the EC keys are sent almost twice as fast as the post-quantum keys.

In situations where there is a large bandwidth or low packet loss, we can see that the effect of the size of the data doesn't play as big a role as it did before.

# 12. Certificate Size vs Write Speed

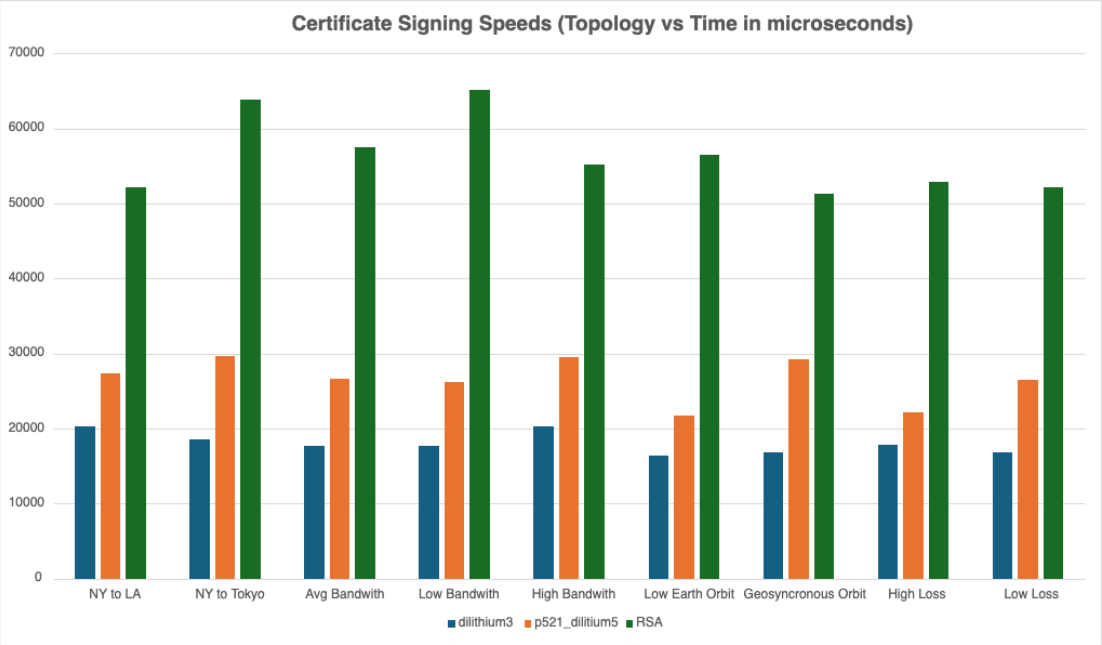


**Figure 12.1** Certificate Signing Speed Results

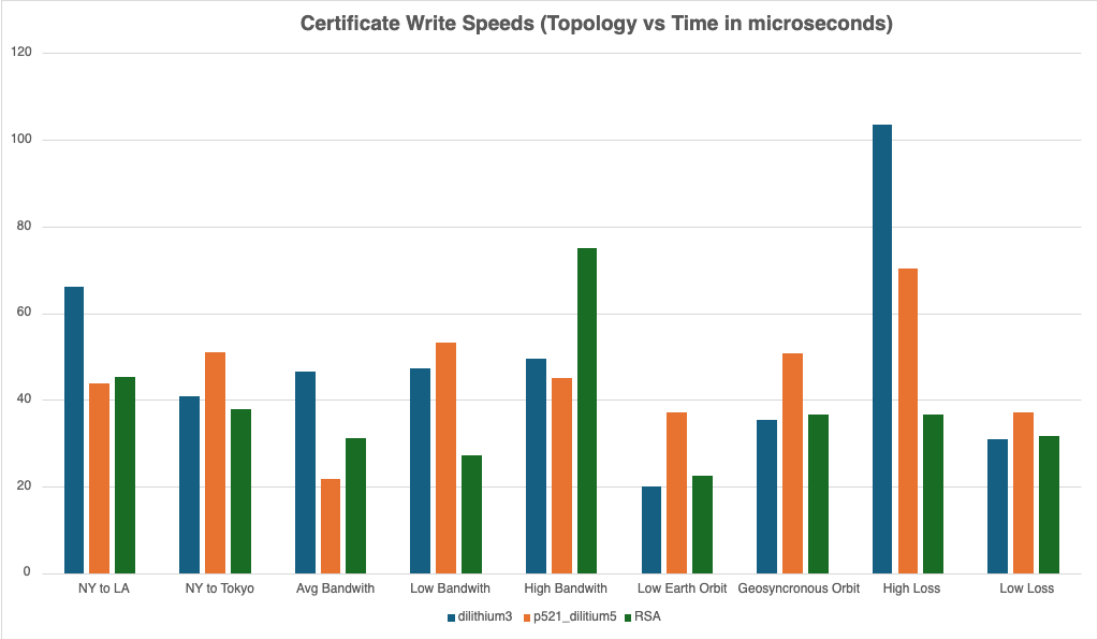


**Figure 12.2** Certificate Writing Speed Results

It came as a surprise to us that the time it takes to generate and sign an RSA certificate is much longer than the time it takes to generate and sign post-quantum certificates. This goes against the intuition because RSA certificates are much smaller than Dilithium3 certificates and P521_Dilithium5 certificates.

However, the nature of RSA and the way it uses large prime numbers to generate these certificates uses more computation time than the post-quantum algorithms. So, even though a smaller certificate is being produced, it takes longer to make it.

On the other hand, the small certificate size of RSA works to make it much easier and quicker to transfer the certificates from one host to the next. So, in situations like High Packet Loss, the transfer is almost twice as fast as the post-quantum transfer. This effect is not felt as much in more ideal situations like low packet loss or low latency.

# 13. Discussion

After running these different test cases and gathering information, there are many different ideas that can be taken away.

Firstly, on average, in ideal situations, post-quantum cryptography does not pose much of a threat to performance in a network. The increased size in cryptographic elements like certificates and keys is noticeable, but when there is sufficient bandwidth to handle this, the transfer of the data is not out of proportion.

In less ideal situations, the larger sizes of the components can play a larger role in slowing down the performance of the network. This can quickly become more catastrophic when the application using cryptography is very reliant on generating new keys or new certificates very frequently. Applications like OpenZiti generate keys and certificates much more often than our program does (we only do each once in the lifetime of the program). Therefore, if you are in a network that is heavily congested, or if you are communicating with a geosynchronous satellite, and you have a program that is constantly generating new keys and new certificates, the effects of post-quantum cryptography will be much more noticeable.

Looking ahead, there is much more to be done in order to gather as much information as possible. Given unlimited time and resources, one thing we would love to look more into is gathering a larger sample size than 5 for every permutation of algorithms at each topology. This would help round out the different times and make sure that small outliers do not skew the overall data as much.

Along with this, we would love to test more client/server applications on these topologies. If we were to implement post-quantum cryptography on an application that relies more on generating, verifying, and transferring cryptographic elements, then the results we have featured in this document would be magnified at a larger scale.