April, 2024
High-Performance Computing

Cryptology & High-Performance Computing

Cassandra Haydock

Introduction

The field of cryptology has long relied on complex mathematical operations for securing sensitive information and communication channels. With the ever-increasing demand for robust encryption techniques, the role of High-Performance Computing (HPC) in cryptology has become indispensable. HPC systems offer the computational power necessary to parallelize complex mathematical operations, thereby significantly enhancing the performance of cryptographic algorithms.

Among classical ciphers, the Vigenère cipher stands as a notable example of early cryptographic techniques. While not as secure as modern encryption algorithms, the Vigenère cipher provides an excellent platform for exploring the application of HPC techniques in cryptology. By parallelizing the complex mathematical operations involved in the Vigenère cipher, I aim to demonstrate how HPC can improve the efficiency and scalability of classical encryption techniques.

High-Performance Computing & it's Application in Cryptology

As encryption techniques evolve to tackle modern cyber threats, they also become computationally intensive. Algorithms like RSA, ECC (Elliptic Curve Cryptography), and AES (Advanced Encryption Standard) require significant computing power for encryption, decryption, and key generation. High-performance computing systems provide the necessary computational power to carry out these tasks efficiently.

One of the primary reasons for the indispensability of HPC in cryptology is its role in combating brute force attacks. Brute force attacks involve trying all possible combinations of keys until the correct one is found. With the increasing processing power of modern computers, traditional cryptographic methods can be vulnerable to brute force attacks. HPC systems enable cryptographers to test encryption algorithms against these attacks by simulating massive computational power (Joux, 2021).

HPC is also instrumental in cryptanalysis, the science of breaking cryptographic systems. Cryptanalysts use many techniques such as mathematical analysis, statistical analysis, and computational methods to find weaknesses or vulnerabilities in encryption algorithms. HPC systems accelerate the cryptanalysis process by allowing for rapid testing and analysis of large datasets and complex mathematical operations.

The emergence of quantum computing poses a significant threat to traditional cryptographic methods. Quantum computers have the potential to solve certain mathematical problems, such as integer factorization and discrete logarithms (which underpin many encryption algorithms), much faster than classical computers (Arel, 2023). For instance, quantum computers can factor large numbers exponentially faster, a crucial capability for breaking classical encryption schemes like RSA. This means there is an increasing need for robust encryption techniques to protect sensitive data in industries like healthcare, finance, and the government. HPC is crucial for this, as it is needed to develop quantum-resistant cryptographic algorithms and protocols to counter the threat posed by quantum computing. For example, NVIDIA announced a project they are starting where they will be using two supercomputer centres to create large quantum emulation capabilities. One centre will be used to research new quantum algorithms as well as techniques for integrating classical and quantum computers in a hybrid configuration. The size of this installation will make it one of the most powerful systems in the world for using classical technology to emulate quantum processors (Finke, 2024).

HPC allows for the implementation of encryption solutions capable of protecting data against sophisticated cyber-attacks and unauthorized access. In applications requiring real-time encryption and decryption, such as secure communication channels and data transmission, the distributed systems and computing power of HPC ensure timely processing of cryptographic operations.

HPC Power & Techniques

Continuing from the last discussion about HPC's application in cryptography, you may be asking, what components of HPC help with cryptography? I'm glad you asked.

In cryptology, parallelization can be applied to encryption and decryption algorithms to speed up computation. For example, symmetric key encryption algorithms like AES can benefit from parallelization techniques such as parallel substitution-permutation network (SPN) rounds or parallel processing of multiple blocks of data simultaneously (Mochurd & Shcur, 2021). SPN for example is a series of linked transformations that take a block of the plaintext as inputs and apply several "rounds" of substitution (S-boxes) and Permutation (P-boxes) (Wolfram U, 2023). Substitution implements confusion, where it attempts to make the relationship between the key and the ciphertext as complex as possible. Diffusion is implemented in Permutation to spread out or rearrange bits in the message so that any redundancy in the plaintext is spread out over the ciphertext (Wolfram U, 2023). Each round applies a substitution and permutation to the input, and there are multiple rounds to make the mapping of input and outputs more complicated. For each round a key is derived from the initial secret key and added to the ciphertext (Wolfram U, 2023).

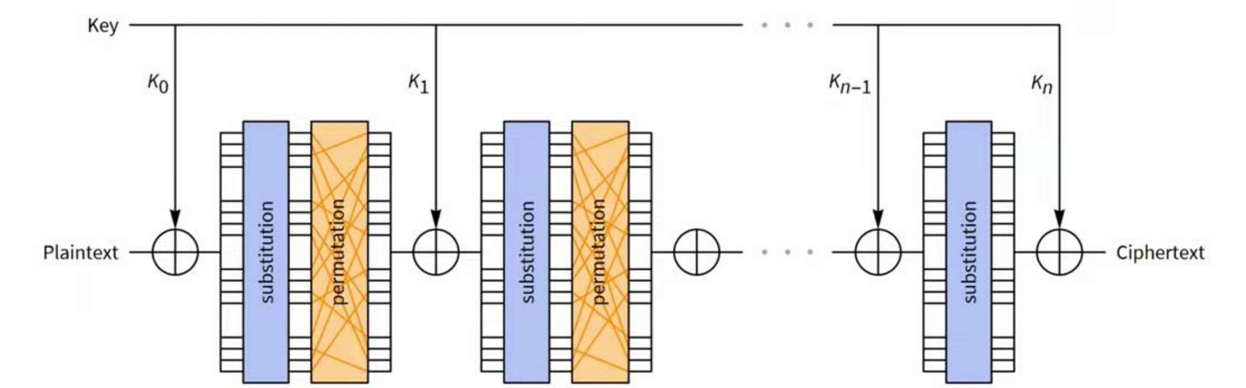


Figure 1: SPN network flow, from Wolfram U, 2023

In SPNs, substitution involves replacing input data with output data based on a substitution table or S-box. This operation is typically applied independently to each block of data. Parallelization can be achieved by processing multiple blocks of data simultaneously, utilizing multiple processing units or SIMD (Single Instruction, Multiple Data) instructions in modern processors. This allows for simultaneous application of substitution across multiple data blocks, increasing throughput and efficiency (Wolfram U, 2023). Permutation involves rearranging the order of bits within each block of data according to a predefined permutation rule. Similar to substitution, permutation can be parallelized by processing multiple blocks of data simultaneously. SIMD instructions or parallel processing units can be utilized to perform permutation operations in parallel (Wolfram U, 2023). Also, each round operates independently on different parts of the data. This means that the operations within a round can be parallelized without dependencies on the results of other rounds. Parallelizing SPNs enable encryption of multiple data blocks simultaneously, leading to a significant increase in throughput.

Cryptographic algorithms also often involve the generation of large prime numbers or random keys, which can be computationally intensive tasks. HPC systems can parallelize the generation of these keys across multiple processing units, thereby speeding up the process. This parallelism is especially beneficial in asymmetric encryption algorithms like RSA, where key generation involves complex mathematical operations such as modular exponentiation (Ayub, et, al., 2019). This parallelism is crucial for tasks such as integer factorization (used in breaking RSA encryption) and discrete logarithm computation (used in breaking Diffie-Hellman key exchange).

Cryptography can use HPC's distributed computing to distribute the computational load of encryption and decryption processes across multiple nodes in a network. This is particularly useful for tasks that involve processing large amounts of data or conducting brute-force attacks. Distributed computing frameworks like Apache Hadoop or Spark can be leveraged for parallelizing cryptanalysis tasks across a cluster of machines.

In addition to general-purpose processors, HPC systems often incorporate specialized hardware accelerators optimized for cryptographic operations. These accelerators, such as cryptographic co-processors or GPUs with dedicated cryptographic instructions, are designed to efficiently execute the specific mathematical operations required by cryptographic algorithms. More and more applications are requiring cryptographic solutions to protect their data; therefore, hardware manufacturers have suggested implementing the popular cryptographic primitives directly onto the hardware. The advantage of this is lower latency for the operations, higher throughput for large transactions, as well as lower overall power consumption (Bloom & Simha, 2012). The challenge here is to construct implementations that offer the most throughput, in the smallest amount of space with good latency and power. Examples of possible implementations are below (Bloom & Simha, 2012).

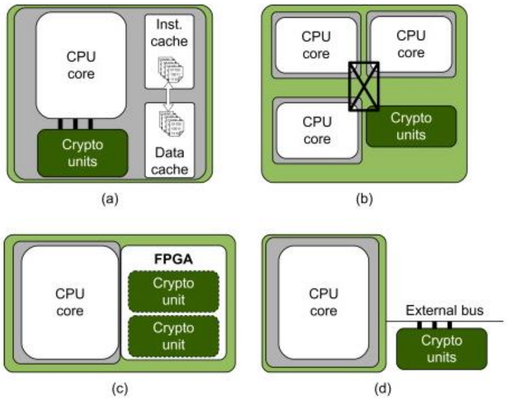


Figure 2: Configurations for connecting a cryptographic accelerator to a CPU: (a) tightly coupled to the pipeline, (b) attached over internal interconnect, (c) synthesized in specialized reconfigurable logic, and (d) attached as a coprocessor. From Bloom & Simha, 2012)

Certain processors are designed specifically to optimize cryptographic tasks, so they feature specialized resources within their execution units that are dedicated to handling tasks like key storage, exponentiation, and modulus arithmetic. These optimized processors streamline cryptographic operations directly in their instruction sets (Bloom & Simha, 2012). However, they rely on compiler support to generate optimized code, rather than standalone cryptographic algorithms. Also, this approach still strains the processor pipeline for cryptographic operations. To combat this, different strategies exist for implementing cryptographic accelerators. These include small, dedicated cores for specific functions, versatile cryptographic coprocessors capable of handling diverse operations, or general-purpose cores designated for particular cryptographic algorithms (Bloom & Simha, 2012). A common integration method involves incorporating the cryptographic engine as a coprocessor. This arrangement connects seamlessly to the system via standard interfaces, allowing the main processor to delegate specific tasks to the coprocessor. Through direct memory access (DMA), the coprocessor can access data in memory (Bloom & Simha, 2012).

Cryptography for HPC

I have discussed how HPC helps protect security and privacy through cryptography, but how about how cryptography helps protect HPC?

Cybersecurity plays a crucial role in High Performance Computing (HPC) because it's vital to safeguard the data and intellectual property used and created by these powerful computing systems, ensuring their protection, availability, confidentiality, and authenticity. One big issue facing High-Performance Computing (HPC) is dealing with a large, and continuously growing, amount of data that needs to be processed really fast. This means data has to be encrypted and decrypted quickly and efficiently to keep it safe. To manage the security challenge in HPC, Secure IC (2022) suggests using fully homomorphic encryption. This fancy method lets you do calculations on encrypted data without having to decrypt it first. This saves a ton of time and lets you process sensitive info from anywhere without worrying about exposing it. It's especially handy for heavy-duty computing tasks like those in Cloud computing, where the computing power on hand might not be enough. By using fully homomorphic encryption in the HPC setup or in the Cloud, you can securely hand off big data processing to third parties you might not fully trust. This makes it possible to handle large-scale data processing securely without putting your sensitive info at risk (Secure IC, 2022).
Secure IC (2022) also discusses using a cryptographic library to deploy solutions to provide countermeasures against a wide range of attacks like cache-timing attacks, and more. The library solution would have embedded in it, multiple software implementations of cryptographic algorithms such as AES, RSA-based cryptography, ECC-based cryptography and hash and MAC functions.

An example of cryptography being used to protect HPC is Google's crusade to encrypt all the data the comes through their data centres in order to protect against government spying. This move comes as a reaction to the leaked Edward Snowden documents that talked about the NSA's project PRISM which gathered information from American tech giants like Google. It was said that NSA tries to defeat encryption through many ways, like leveraging many, super expensive supercomputers to break codes, and by influencing encryption standards to make them more vulnerable from outside attacks (Timberg, 2013). The information traveling between companies like Googles data centers offered rare points of vulnerability to potential intruders, which the government surveillance agencies could use to obtain data. User information — including copies of e-mails, search queries, videos and Web browsing history — typically is stored in several data centers that transmit information to each other on high-speed fiber-optic lines (Timberg, 2013).

Google encrypts data once it is stored on its server or a back up medium (encryption at rest) as well as when the data is in transit (encryption in transit) (Google Cloud Tec, 2021). There are also many layers of encryption used. The data is split into chunks and each chunk is encrypted with a unique encryption key. These keys are stored with the data and encrypted inside of googles central key management service. Google key management service rotates keys at least as frequent as 90 days, so if a key were to become compromised it won't remain valid for very long (Google Cloud Tec, 2021). For example, if a video was uploaded to google drive, the entire video will not be stored on a single server, it will be split into chunks and then copied for availability and performance. Each chunk is encrypted and have their own unique ID, and each piece is then stored across multiple servers. As for when the data is in transit, google applies default protection to data in transit to a google data centre. For example, communication between the user and the google front end is protected using TLS which is a cryptographic protocol. Data is encrypted and authenticated in transit at one or more network layers when it moves outside the physical boundaries of a Google data center (Google Cloud Tec, 2021).

Vigenère Cipher Parallelization

The Vigenère Cipher is a way to encode text with letters. It's like a simple version of mixing up letters in different ways. Basically, it's a type of code that uses different ways of substituting letters. When you want to encrypt a message, you can use this thing called the Vigenère square or table to do it (Geeks for Geeks, 2023).

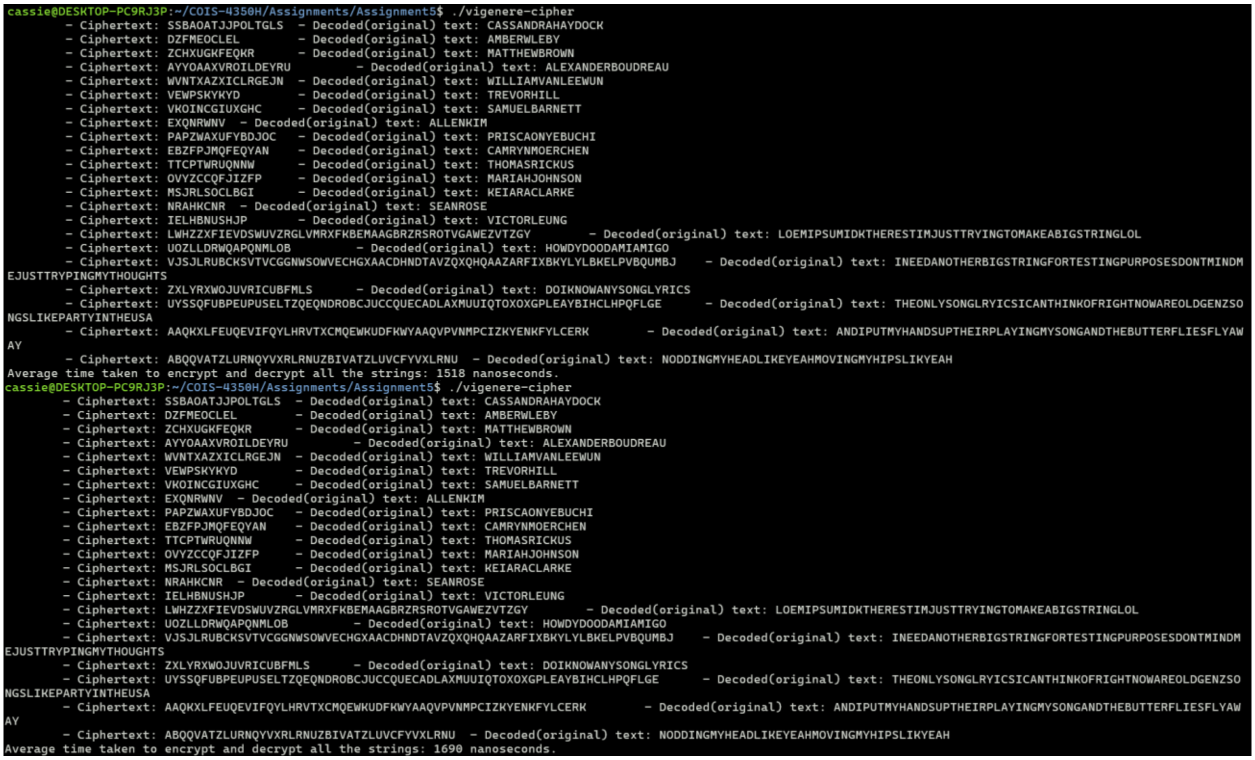
Parallelizing a simpler encryption algorithm like the Vigenère cipher demonstrates the efficacy of parallelization in cryptology by showcasing how computational resources can be leveraged to enhance encryption and decryption processes, regardless of algorithm complexity. While the Vigenère cipher may be simpler compared to modern cryptographic techniques, its parallelization underscores the fundamental principle that breaking down computational tasks into concurrent operations can significantly expedite cryptographic operations. By distributing the workload across multiple threads or processing units, parallelization minimizes processing time, making encryption and decryption faster and more efficient. This optimization is particularly relevant in scenarios where large volumes of data need to be encrypted or decrypted within stringent time constraints. Moreover, the parallelization of simpler algorithms like the Vigenère cipher serves as a foundational demonstration of how parallel computing principles can be applied to more complex cryptographic systems, highlighting the broader applicability of parallelization techniques in advancing cryptology as a whole.

I have taken a Vigenère Cipher implementation from Geeks for Geeks (2023) and have modified it to encrypt and decrypt multiple strings, each with a unique key. After making the modifications, I parallelized the implementation using MPI. The logic is that the code divides the work of encrypting and decrypting all the strings/keys between all the processes (i.e. chunk size = number of string to encrypt / number of processes). Each process encrypts and decrypt a certain number of strings, times how long it takes for each string, adds that to a total time variable (called result), then I use MPI_Barrier to make all processes wait until they are done before I use MPI_Reduce to add all the times together to get the average time for the entire ordeal. I got the most amount of speedup, faster than the non-parallelized version, when using 2 processes.

Example Output

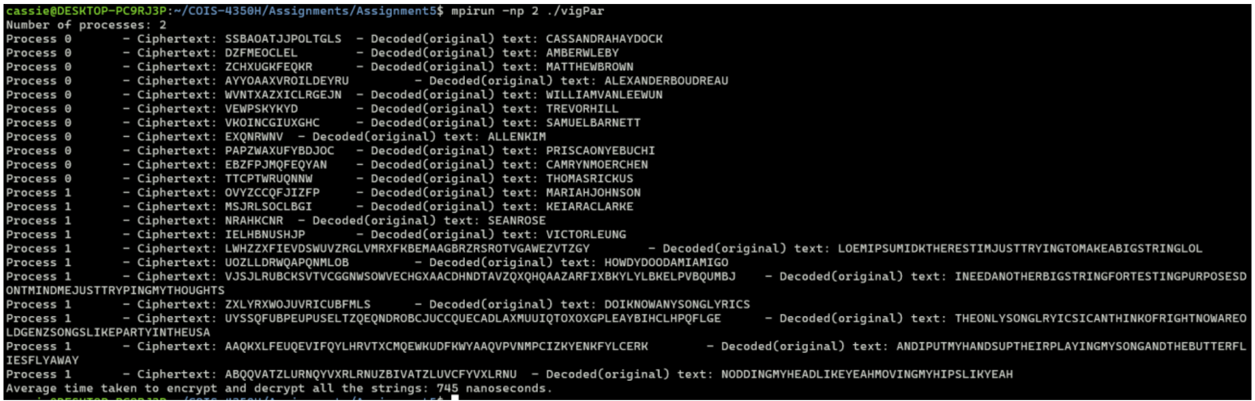
22 entries

Non-Parallelized

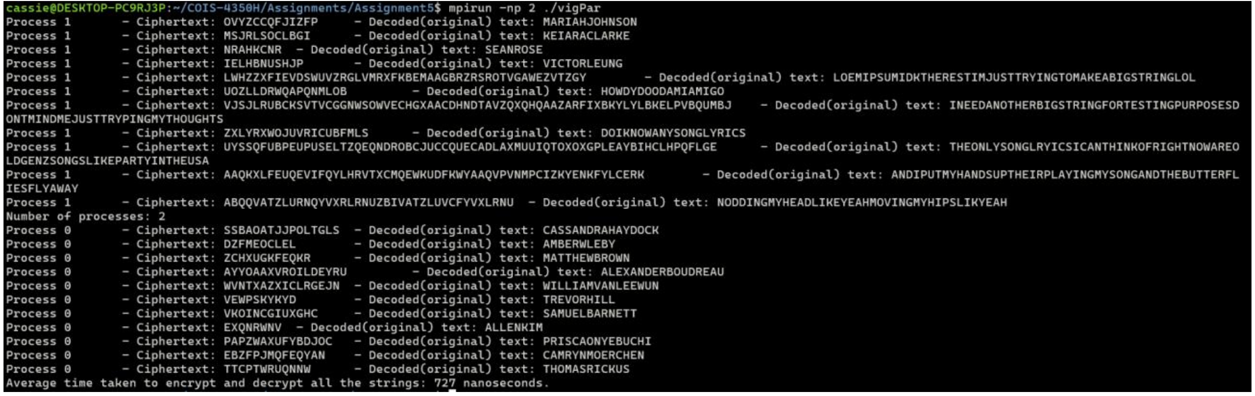


```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ ./vigenere-cipher
        - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
        - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
        - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
        - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
        - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
        - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
        - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
        - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
        - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
        - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
        - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
        - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
        - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
        - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
        - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
        - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
        - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
        - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDM
EJUSTTRYPINGMYTHOUGHTS
        - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
        - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSO
NGSLIKEPARTYINTHEUSA
        - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAW
AY
        - Ciphertext: ABQQVATZLURNQYVXRLRNUZBIVATZLUVCFYVXLRNU  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Average time taken to encrypt and decrypt all the strings: 1518 nanoseconds.
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ ./vigenere-cipher
        - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
        - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
        - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
        - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
        - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
        - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
        - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
        - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
        - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
        - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
        - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
        - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
        - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
        - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
        - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
        - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
        - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
        - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDM
EJUSTTRYPINGMYTHOUGHTS
        - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
        - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSO
NGSLIKEPARTYINTHEUSA
        - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAW
AY
        - Ciphertext: ABQQVATZLURNQYVXRLRNUZBIVATZLUVCFYVXLRNU  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Average time taken to encrypt and decrypt all the strings: 1690 nanoseconds.
```

Parallelized



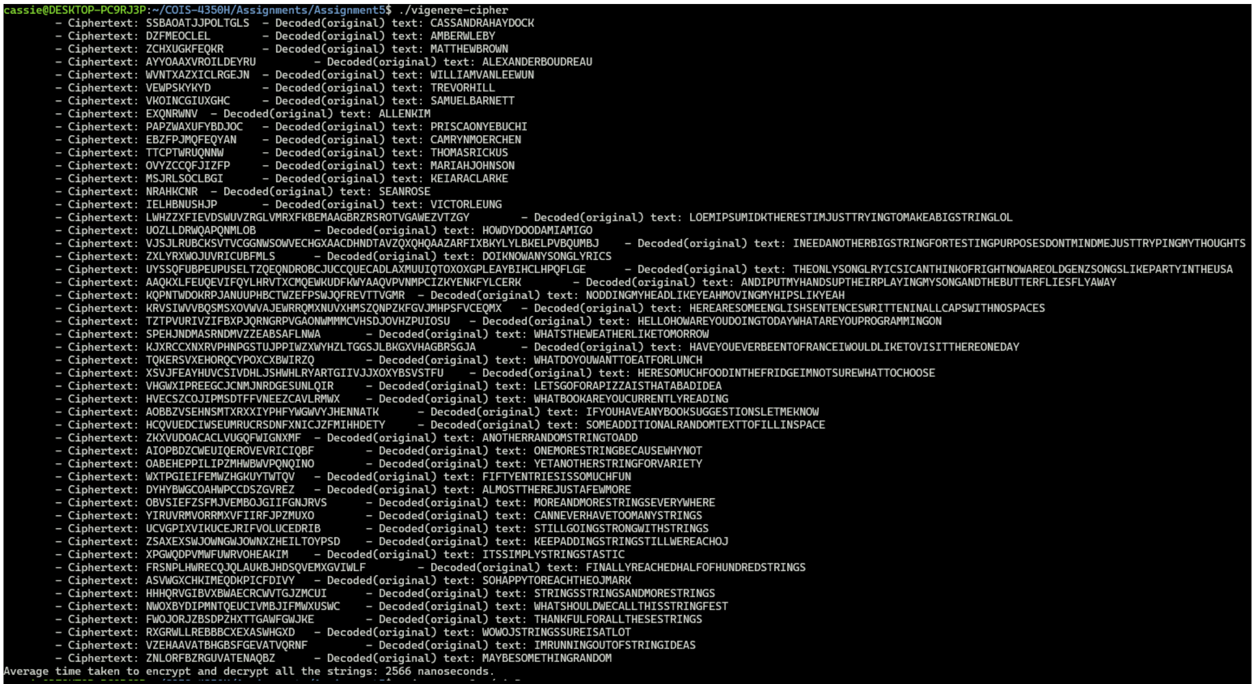
```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ mpirun -np 2 ./vigPar
Number of processes: 2
Process 0       - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
Process 0       - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
Process 0       - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
Process 0       - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
Process 0       - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
Process 0       - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
Process 0       - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
Process 0       - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
Process 0       - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
Process 0       - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
Process 0       - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
Process 1       - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
Process 1       - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
Process 1       - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
Process 1       - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
Process 1       - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
Process 1       - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
Process 1       - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESD
ONTMINDMEJUSTTRYPINGMYTHOUGHTS
Process 1       - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
Process 1       - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREO
LDGENZSONGSLIKEPARTYINTHEUSA
Process 1       - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFL
IESFLYAWAY
Process 1       - Ciphertext: ABQQVATZLURNQYVXRLRNUZBIVATZLUVCFYVXLRNU  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Average time taken to encrypt and decrypt all the strings: 745 nanoseconds.
```



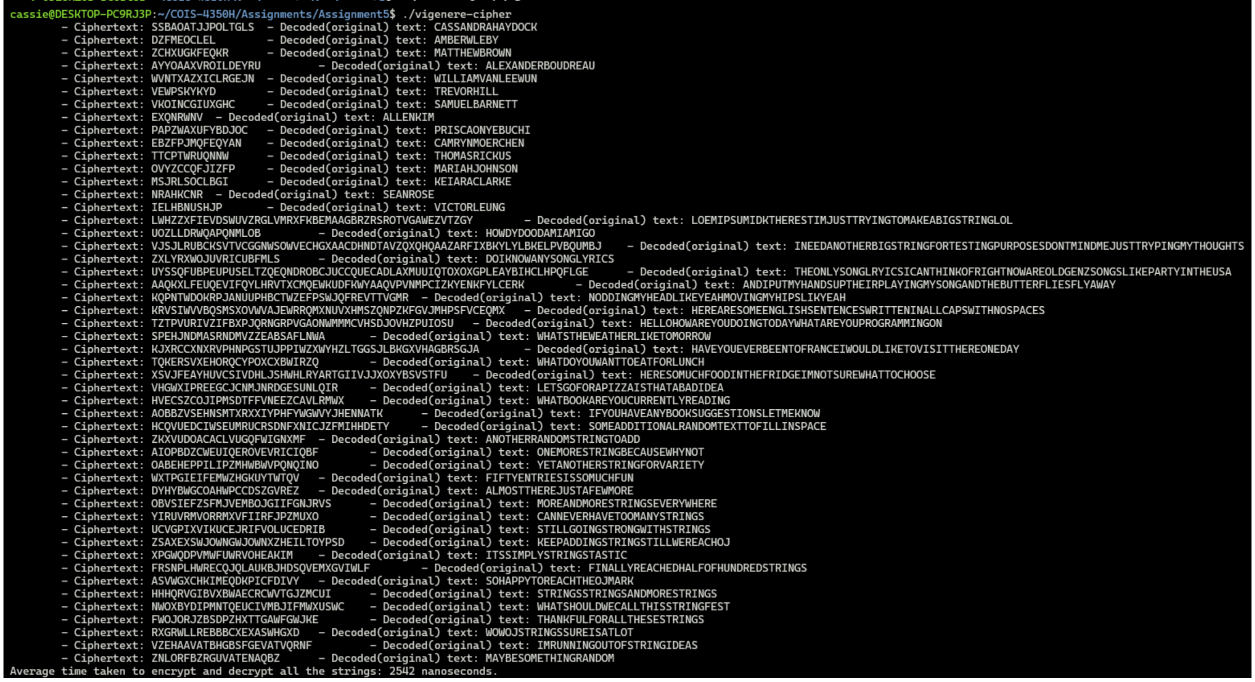
```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ mpirun -np 2 ./vigPar
Process 1       - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
Process 1       - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
Process 1       - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
Process 1       - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
Process 1       - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
Process 1       - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
Process 1       - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESD
ONTMINDMEJUSTTRYPINGMYTHOUGHTS
Process 1       - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
Process 1       - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREO
LDGENZSONGSLIKEPARTYINTHEUSA
Process 1       - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFL
IESFLYAWAY
Process 1       - Ciphertext: ABQQVATZLURNQYVXRLRNUZBIVATZLUVCFYVXLRNU  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Number of processes: 2
Process 0       - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
Process 0       - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
Process 0       - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
Process 0       - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
Process 0       - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
Process 0       - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
Process 0       - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
Process 0       - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
Process 0       - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
Process 0       - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
Process 0       - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
Average time taken to encrypt and decrypt all the strings: 727 nanoseconds.
```

50 entries

Non-Parallelized

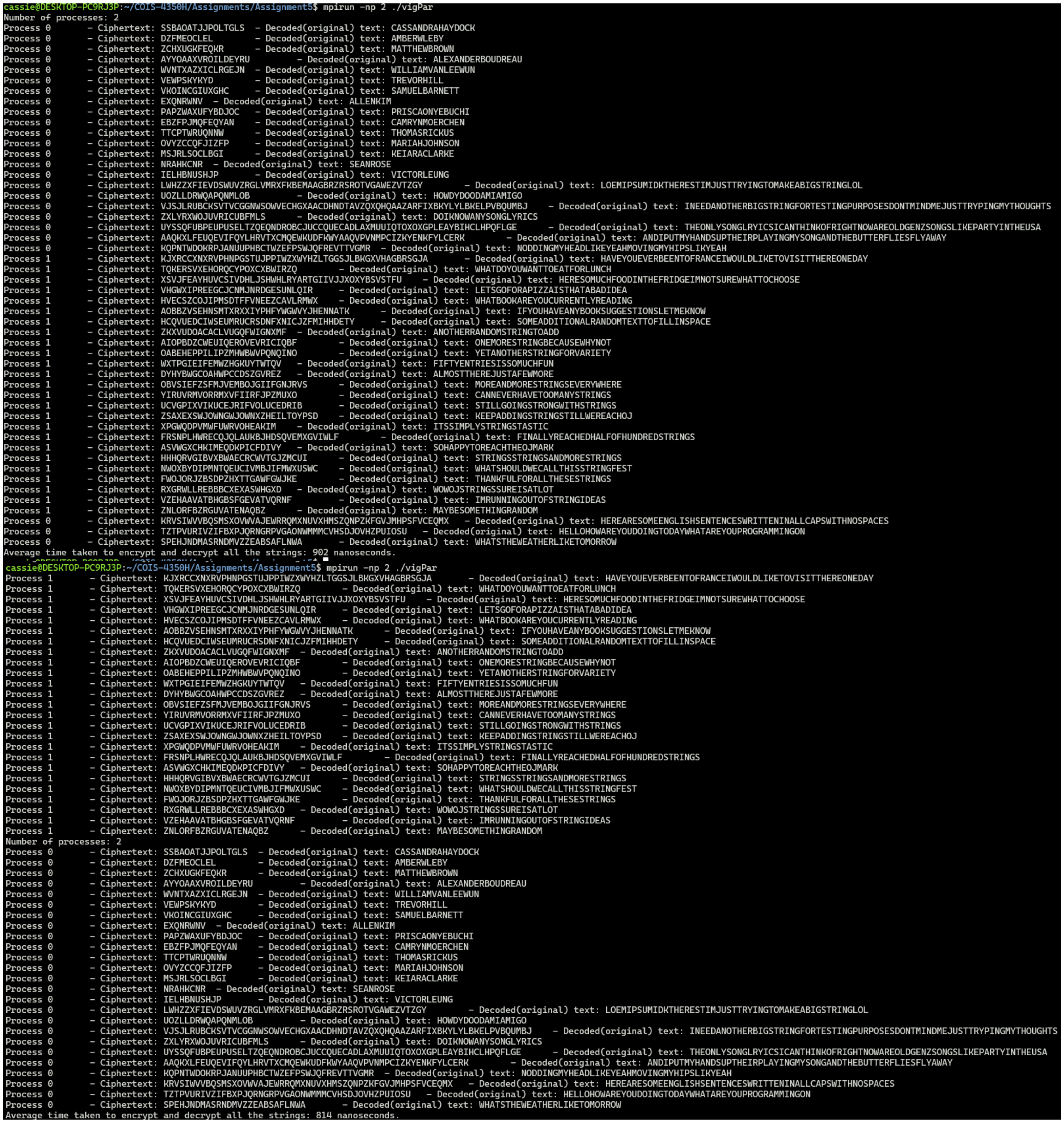


```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ ./vigenere-cipher
        - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
        - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
        - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
        - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
        - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
        - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
        - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
        - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
        - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
        - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
        - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
        - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
        - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
        - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
        - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
        - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
        - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
        - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDMEJUSTTRYPINGMYTHOUGHTS
        - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
        - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSONGSLIKEPARTYINTHEUSA
        - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAWAY
        - Ciphertext: KQPNTWDOKRPJANUUPHBCTWZEFPSWJQFREVTTVGMR  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
        - Ciphertext: KRVSIWVVBQSMSXOVWVAJEWRRQMXNUVXHMSZQNPZKFGVJMHPSFVCEQMX   - Decoded(original) text: HEREARESOMEENGLISHSENTENCESWRITTENINALLCAPSWITHNOSPACES
        - Ciphertext: TZTPVURIVZIFBXPJQRNGRPVGAONWMMMCVHSDJOVHZPUIOSU   - Decoded(original) text: HELLOHOWAREYOUDOINGTODAYWHATAREYOUPROGRAMMINGON
        - Ciphertext: SPEHJNDMASRNDMVZZEABSAFLNWA       - Decoded(original) text: WHATSTHEWEATHERLIKETOMORROW
        - Ciphertext: KJXRCCXNXRVPHNPGSTUJPPIWZXWYHZLTGGSJLBKGXVHAGBRSGJA       - Decoded(original) text: HAVEYOUEVERBEENTOFRANCEIWOULDLIKETOVISITTHEREONEDAY
        - Ciphertext: TQKERSVXEHORQCYPOXCXBWIRZQ        - Decoded(original) text: WHATDOYOUWANTTOEATFORLUNCH
        - Ciphertext: XSVJFEAYHUVCSIVDHLJSHWHLRYARTGIIVJJXOXYBSVSTFU    - Decoded(original) text: HERESOMUCHFOODINTHEFRIDGEIMNOTSUREWHATTOCHOOSE
        - Ciphertext: VHGWXIPREEGCJCNMJNRDGESUNLQIR     - Decoded(original) text: LETSGOFORAPIZZAISTHATABADIDEA
        - Ciphertext: HVECSZCOJIPMSDTFFVNEEZCAVLRMWX    - Decoded(original) text: WHATBOOKAREYOUCURRENTLYREADING
        - Ciphertext: AOBBZVSEHNSMTXRXXIYPHFYWGWVYJHENNATK      - Decoded(original) text: IFYOUHAVEANYBOOKSUGGESTIONSLETMEKNOW
        - Ciphertext: HCQVUEDCIWSEUMRUCRSDNFXNICJZFMIHHDETY     - Decoded(original) text: SOMEADDITIONALRANDOMTEXTTOFILLINSPACE
        - Ciphertext: ZKXVUDOACACLVUGQFWIGNXMF  - Decoded(original) text: ANOTHERRANDOMSTRINGTOADD
        - Ciphertext: AIOPBDZCWEUIQEROVEVRICIQBF        - Decoded(original) text: ONEMORESTRINGBECAUSEWHYNOT
        - Ciphertext: OABEHEPPILIPZMHWBWVPQNQINO        - Decoded(original) text: YETANOTHERSTRINGFORVARIETY
        - Ciphertext: WXTPGIEIFEMWZHGKUYTWTQV   - Decoded(original) text: FIFTYENTRIESISSOMUCHFUN
        - Ciphertext: DYHYBWGCOAHWPCCDSZGVREZ   - Decoded(original) text: ALMOSTTHEREJUSTAFEWMORE
        - Ciphertext: OBVSIEFZSFMJVEMBOJGIIFGNJRVS      - Decoded(original) text: MOREANDMORESTRINGSEVERYWHERE
        - Ciphertext: YIRUVRMVORRMXVFIIRFJPZMUXO        - Decoded(original) text: CANNEVERHAVETOOMANYSTRINGS
        - Ciphertext: UCVGPIXVIKUCEJRIFVOLUCEDRIB       - Decoded(original) text: STILLGOINGSTRONGWITHSTRINGS
        - Ciphertext: ZSAXEXSWJOWNGWJOWNXZHEILTOYPSD    - Decoded(original) text: KEEPADDINGSTRINGSTILLWEREACHOJ
        - Ciphertext: XPGWQDPVMWFUWRVOHEAKIM    - Decoded(original) text: ITSSIMPLYSTRINGSTASTIC
        - Ciphertext: FRSNPLHWRECQJQLAUKBJHDSQVEMXGVIWLF        - Decoded(original) text: FINALLYREACHEDHALFOFHUNDREDSTRINGS
        - Ciphertext: ASVWGXCHKIMEQDKPICFDIVY   - Decoded(original) text: SOHAPPYTOREACHTHEOJMARK
        - Ciphertext: HHHQRVGIBVXBWAECRCWVTGJZMCUI      - Decoded(original) text: STRINGSSTRINGSANDMORESTRINGS
        - Ciphertext: NWOXBYDIPMNTQEUCIVMBJIFMWXUSWC    - Decoded(original) text: WHATSHOULDWECALLTHISSTRINGFEST
        - Ciphertext: FWOJORJZBSDPZHXTTGAWFGWJKE        - Decoded(original) text: THANKFULFORALLTHESESTRINGS
        - Ciphertext: RXGRWLLREBBBCXEXASWHGXD   - Decoded(original) text: WOWOJSTRINGSSUREISATLOT
        - Ciphertext: VZEHAAVATBHGBSFGEVATVQRNF         - Decoded(original) text: IMRUNNINGOUTOFSTRINGIDEAS
        - Ciphertext: ZNLORFBZRGUVATENAQBZ      - Decoded(original) text: MAYBESOMETHINGRANDOM
Average time taken to encrypt and decrypt all the strings: 2566 nanoseconds.
```



```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ ./vigenere-cipher
        - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
        - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
        - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
        - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
        - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
        - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
        - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
        - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
        - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
        - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
        - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
        - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
        - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
        - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
        - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
        - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
        - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
        - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDMEJUSTTRYPINGMYTHOUGHTS
        - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
        - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSONGSLIKEPARTYINTHEUSA
        - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAWAY
        - Ciphertext: KQPNTWDOKRPJANUUPHBCTWZEFPSWJQFREVTTVGMR  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
        - Ciphertext: KRVSIWVVBQSMSXOVWVAJEWRRQMXNUVXHMSZQNPZKFGVJMHPSFVCEQMX   - Decoded(original) text: HEREARESOMEENGLISHSENTENCESWRITTENINALLCAPSWITHNOSPACES
        - Ciphertext: TZTPVURIVZIFBXPJQRNGRPVGAONWMMMCVHSDJOVHZPUIOSU   - Decoded(original) text: HELLOHOWAREYOUDOINGTODAYWHATAREYOUPROGRAMMINGON
        - Ciphertext: SPEHJNDMASRNDMVZZEABSAFLNWA       - Decoded(original) text: WHATSTHEWEATHERLIKETOMORROW
        - Ciphertext: KJXRCCXNXRVPHNPGSTUJPPIWZXWYHZLTGGSJLBKGXVHAGBRSGJA       - Decoded(original) text: HAVEYOUEVERBEENTOFRANCEIWOULDLIKETOVISITTHEREONEDAY
        - Ciphertext: TQKERSVXEHORQCYPOXCXBWIRZQ        - Decoded(original) text: WHATDOYOUWANTTOEATFORLUNCH
        - Ciphertext: XSVJFEAYHUVCSIVDHLJSHWHLRYARTGIIVJJXOXYBSVSTFU    - Decoded(original) text: HERESOMUCHFOODINTHEFRIDGEIMNOTSUREWHATTOCHOOSE
        - Ciphertext: VHGWXIPREEGCJCNMJNRDGESUNLQIR     - Decoded(original) text: LETSGOFORAPIZZAISTHATABADIDEA
        - Ciphertext: HVECSZCOJIPMSDTFFVNEEZCAVLRMWX    - Decoded(original) text: WHATBOOKAREYOUCURRENTLYREADING
        - Ciphertext: AOBBZVSEHNSMTXRXXIYPHFYWGWVYJHENNATK      - Decoded(original) text: IFYOUHAVEANYBOOKSUGGESTIONSLETMEKNOW
        - Ciphertext: HCQVUEDCIWSEUMRUCRSDNFXNICJZFMIHHDETY     - Decoded(original) text: SOMEADDITIONALRANDOMTEXTTOFILLINSPACE
        - Ciphertext: ZKXVUDOACACLVUGQFWIGNXMF  - Decoded(original) text: ANOTHERRANDOMSTRINGTOADD
        - Ciphertext: AIOPBDZCWEUIQEROVEVRICIQBF        - Decoded(original) text: ONEMORESTRINGBECAUSEWHYNOT
        - Ciphertext: OABEHEPPILIPZMHWBWVPQNQINO        - Decoded(original) text: YETANOTHERSTRINGFORVARIETY
        - Ciphertext: WXTPGIEIFEMWZHGKUYTWTQV   - Decoded(original) text: FIFTYENTRIESISSOMUCHFUN
        - Ciphertext: DYHYBWGCOAHWPCCDSZGVREZ   - Decoded(original) text: ALMOSTTHEREJUSTAFEWMORE
        - Ciphertext: OBVSIEFZSFMJVEMBOJGIIFGNJRVS      - Decoded(original) text: MOREANDMORESTRINGSEVERYWHERE
        - Ciphertext: YIRUVRMVORRMXVFIIRFJPZMUXO        - Decoded(original) text: CANNEVERHAVETOOMANYSTRINGS
        - Ciphertext: UCVGPIXVIKUCEJRIFVOLUCEDRIB       - Decoded(original) text: STILLGOINGSTRONGWITHSTRINGS
        - Ciphertext: ZSAXEXSWJOWNGWJOWNXZHEILTOYPSD    - Decoded(original) text: KEEPADDINGSTRINGSTILLWEREACHOJ
        - Ciphertext: XPGWQDPVMWFUWRVOHEAKIM    - Decoded(original) text: ITSSIMPLYSTRINGSTASTIC
        - Ciphertext: FRSNPLHWRECQJQLAUKBJHDSQVEMXGVIWLF        - Decoded(original) text: FINALLYREACHEDHALFOFHUNDREDSTRINGS
        - Ciphertext: ASVWGXCHKIMEQDKPICFDIVY   - Decoded(original) text: SOHAPPYTOREACHTHEOJMARK
        - Ciphertext: HHHQRVGIBVXBWAECRCWVTGJZMCUI      - Decoded(original) text: STRINGSSTRINGSANDMORESTRINGS
        - Ciphertext: NWOXBYDIPMNTQEUCIVMBJIFMWXUSWC    - Decoded(original) text: WHATSHOULDWECALLTHISSTRINGFEST
        - Ciphertext: FWOJORJZBSDPZHXTTGAWFGWJKE        - Decoded(original) text: THANKFULFORALLTHESESTRINGS
        - Ciphertext: RXGRWLLREBBBCXEXASWHGXD   - Decoded(original) text: WOWOJSTRINGSSUREISATLOT
        - Ciphertext: VZEHAAVATBHGBSFGEVATVQRNF         - Decoded(original) text: IMRUNNINGOUTOFSTRINGIDEAS
        - Ciphertext: ZNLORFBZRGUVATENAQBZ      - Decoded(original) text: MAYBESOMETHINGRANDOM
Average time taken to encrypt and decrypt all the strings: 2542 nanoseconds.
```

Parallelized



```
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ mpirun -np 2 ./vigPar
Number of processes: 2
Process 0       - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
Process 0       - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
Process 0       - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
Process 0       - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
Process 0       - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
Process 0       - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
Process 0       - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
Process 0       - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
Process 0       - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
Process 0       - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
Process 0       - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
Process 0       - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
Process 0       - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
Process 0       - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
Process 0       - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
Process 0       - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
Process 0       - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
Process 0       - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDMEJUSTTRYPINGMYTHOUGHTS
Process 0       - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
Process 0       - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSONGSLIKEPARTYINTHEUSA
Process 0       - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAWAY
Process 0       - Ciphertext: KQPNTWDOKRPJANUUPHBCTWZEFPSWJQFREVTTVGMR  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Process 1       - Ciphertext: KJXRCCXNXRVPHNPGSTUJPPIWZXWYHZLTGGSJLBKGXVHAGBRSGJA       - Decoded(original) text: HAVEYOUEVERBEENTOFRANCEIWOULDLIKETOVISITTHEREONEDAY
Process 1       - Ciphertext: TQKERSVXEHORQCYPOXCXBWIRZQ        - Decoded(original) text: WHATDOYOUWANTTOEATFORLUNCH
Process 1       - Ciphertext: XSVJFEAYHUVCSIVDHLJSHWHLRYARTGIIVJJXOXYBSVSTFU    - Decoded(original) text: HERESOMUCHFOODINTHEFRIDGEIMNOTSUREWHATTOCHOOSE
Process 1       - Ciphertext: VHGWXIPREEGCJCNMJNRDGESUNLQIR     - Decoded(original) text: LETSGOFORAPIZZAISTHATABADIDEA
Process 1       - Ciphertext: HVECSZCOJIPMSDTFFVNEEZCAVLRMWX    - Decoded(original) text: WHATBOOKAREYOUCURRENTLYREADING
Process 1       - Ciphertext: AOBBZVSEHNSMTXRXXIYPHFYWGWVYJHENNATK      - Decoded(original) text: IFYOUHAVEANYBOOKSUGGESTIONSLETMEKNOW
Process 1       - Ciphertext: HCQVUEDCIWSEUMRUCRSDNFXNICJZFMIHHDETY     - Decoded(original) text: SOMEADDITIONALRANDOMTEXTTOFILLINSPACE
Process 1       - Ciphertext: ZKXVUDOACACLVUGQFWIGNXMF  - Decoded(original) text: ANOTHERRANDOMSTRINGTOADD
Process 1       - Ciphertext: AIOPBDZCWEUIQEROVEVRICIQBF        - Decoded(original) text: ONEMORESTRINGBECAUSEWHYNOT
Process 1       - Ciphertext: OABEHEPPILIPZMHWBWVPQNQINO        - Decoded(original) text: YETANOTHERSTRINGFORVARIETY
Process 1       - Ciphertext: WXTPGIEIFEMWZHGKUYTWTQV   - Decoded(original) text: FIFTYENTRIESISSOMUCHFUN
Process 1       - Ciphertext: DYHYBWGCOAHWPCCDSZGVREZ   - Decoded(original) text: ALMOSTTHEREJUSTAFEWMORE
Process 1       - Ciphertext: OBVSIEFZSFMJVEMBOJGIIFGNJRVS      - Decoded(original) text: MOREANDMORESTRINGSEVERYWHERE
Process 1       - Ciphertext: YIRUVRMVORRMXVFIIRFJPZMUXO        - Decoded(original) text: CANNEVERHAVETOOMANYSTRINGS
Process 1       - Ciphertext: UCVGPIXVIKUCEJRIFVOLUCEDRIB       - Decoded(original) text: STILLGOINGSTRONGWITHSTRINGS
Process 1       - Ciphertext: ZSAXEXSWJOWNGWJOWNXZHEILTOYPSD    - Decoded(original) text: KEEPADDINGSTRINGSTILLWEREACHOJ
Process 1       - Ciphertext: XPGWQDPVMWFUWRVOHEAKIM    - Decoded(original) text: ITSSIMPLYSTRINGSTASTIC
Process 1       - Ciphertext: FRSNPLHWRECQJQLAUKBJHDSQVEMXGVIWLF        - Decoded(original) text: FINALLYREACHEDHALFOFHUNDREDSTRINGS
Process 1       - Ciphertext: ASVWGXCHKIMEQDKPICFDIVY   - Decoded(original) text: SOHAPPYTOREACHTHEOJMARK
Process 1       - Ciphertext: HHHQRVGIBVXBWAECRCWVTGJZMCUI      - Decoded(original) text: STRINGSSTRINGSANDMORESTRINGS
Process 1       - Ciphertext: NWOXBYDIPMNTQEUCIVMBJIFMWXUSWC    - Decoded(original) text: WHATSHOULDWECALLTHISSTRINGFEST
Process 1       - Ciphertext: FWOJORJZBSDPZHXTTGAWFGWJKE        - Decoded(original) text: THANKFULFORALLTHESESTRINGS
Process 1       - Ciphertext: RXGRWLLREBBBCXEXASWHGXD   - Decoded(original) text: WOWOJSTRINGSSUREISATLOT
Process 1       - Ciphertext: VZEHAAVATBHGBSFGEVATVQRNF         - Decoded(original) text: IMRUNNINGOUTOFSTRINGIDEAS
Process 1       - Ciphertext: ZNLORFBZRGUVATENAQBZ      - Decoded(original) text: MAYBESOMETHINGRANDOM
Process 0       - Ciphertext: KRVSIWVVBQSMSXOVWVAJEWRRQMXNUVXHMSZQNPZKFGVJMHPSFVCEQMX   - Decoded(original) text: HEREARESOMEENGLISHSENTENCESWRITTENINALLCAPSWITHNOSPACES
Process 0       - Ciphertext: TZTPVURIVZIFBXPJQRNGRPVGAONWMMMCVHSDJOVHZPUIOSU   - Decoded(original) text: HELLOHOWAREYOUDOINGTODAYWHATAREYOUPROGRAMMINGON
Process 0       - Ciphertext: SPEHJNDMASRNDMVZZEABSAFLNWA       - Decoded(original) text: WHATSTHEWEATHERLIKETOMORROW
Average time taken to encrypt and decrypt all the strings: 902 nanoseconds.
cassie@DESKTOP-PC9RJ3P:~/COIS-4350H/Assignments/Assignment5$ mpirun -np 2 ./vigPar
Process 1       - Ciphertext: KJXRCCXNXRVPHNPGSTUJPPIWZXWYHZLTGGSJLBKGXVHAGBRSGJA       - Decoded(original) text: HAVEYOUEVERBEENTOFRANCEIWOULDLIKETOVISITTHEREONEDAY
Process 1       - Ciphertext: TQKERSVXEHORQCYPOXCXBWIRZQ        - Decoded(original) text: WHATDOYOUWANTTOEATFORLUNCH
Process 1       - Ciphertext: XSVJFEAYHUVCSIVDHLJSHWHLRYARTGIIVJJXOXYBSVSTFU    - Decoded(original) text: HERESOMUCHFOODINTHEFRIDGEIMNOTSUREWHATTOCHOOSE
Process 1       - Ciphertext: VHGWXIPREEGCJCNMJNRDGESUNLQIR     - Decoded(original) text: LETSGOFORAPIZZAISTHATABADIDEA
Process 1       - Ciphertext: HVECSZCOJIPMSDTFFVNEEZCAVLRMWX    - Decoded(original) text: WHATBOOKAREYOUCURRENTLYREADING
Process 1       - Ciphertext: AOBBZVSEHNSMTXRXXIYPHFYWGWVYJHENNATK      - Decoded(original) text: IFYOUHAVEANYBOOKSUGGESTIONSLETMEKNOW
Process 1       - Ciphertext: HCQVUEDCIWSEUMRUCRSDNFXNICJZFMIHHDETY     - Decoded(original) text: SOMEADDITIONALRANDOMTEXTTOFILLINSPACE
Process 1       - Ciphertext: ZKXVUDOACACLVUGQFWIGNXMF  - Decoded(original) text: ANOTHERRANDOMSTRINGTOADD
Process 1       - Ciphertext: AIOPBDZCWEUIQEROVEVRICIQBF        - Decoded(original) text: ONEMORESTRINGBECAUSEWHYNOT
Process 1       - Ciphertext: OABEHEPPILIPZMHWBWVPQNQINO        - Decoded(original) text: YETANOTHERSTRINGFORVARIETY
Process 1       - Ciphertext: WXTPGIEIFEMWZHGKUYTWTQV   - Decoded(original) text: FIFTYENTRIESISSOMUCHFUN
Process 1       - Ciphertext: DYHYBWGCOAHWPCCDSZGVREZ   - Decoded(original) text: ALMOSTTHEREJUSTAFEWMORE
Process 1       - Ciphertext: OBVSIEFZSFMJVEMBOJGIIFGNJRVS      - Decoded(original) text: MOREANDMORESTRINGSEVERYWHERE
Process 1       - Ciphertext: YIRUVRMVORRMXVFIIRFJPZMUXO        - Decoded(original) text: CANNEVERHAVETOOMANYSTRINGS
Process 1       - Ciphertext: UCVGPIXVIKUCEJRIFVOLUCEDRIB       - Decoded(original) text: STILLGOINGSTRONGWITHSTRINGS
Process 1       - Ciphertext: ZSAXEXSWJOWNGWJOWNXZHEILTOYPSD    - Decoded(original) text: KEEPADDINGSTRINGSTILLWEREACHOJ
Process 1       - Ciphertext: XPGWQDPVMWFUWRVOHEAKIM    - Decoded(original) text: ITSSIMPLYSTRINGSTASTIC
Process 1       - Ciphertext: FRSNPLHWRECQJQLAUKBJHDSQVEMXGVIWLF        - Decoded(original) text: FINALLYREACHEDHALFOFHUNDREDSTRINGS
Process 1       - Ciphertext: ASVWGXCHKIMEQDKPICFDIVY   - Decoded(original) text: SOHAPPYTOREACHTHEOJMARK
Process 1       - Ciphertext: HHHQRVGIBVXBWAECRCWVTGJZMCUI      - Decoded(original) text: STRINGSSTRINGSANDMORESTRINGS
Process 1       - Ciphertext: NWOXBYDIPMNTQEUCIVMBJIFMWXUSWC    - Decoded(original) text: WHATSHOULDWECALLTHISSTRINGFEST
Process 1       - Ciphertext: FWOJORJZBSDPZHXTTGAWFGWJKE        - Decoded(original) text: THANKFULFORALLTHESESTRINGS
Process 1       - Ciphertext: RXGRWLLREBBBCXEXASWHGXD   - Decoded(original) text: WOWOJSTRINGSSUREISATLOT
Process 1       - Ciphertext: VZEHAAVATBHGBSFGEVATVQRNF         - Decoded(original) text: IMRUNNINGOUTOFSTRINGIDEAS
Process 1       - Ciphertext: ZNLORFBZRGUVATENAQBZ      - Decoded(original) text: MAYBESOMETHINGRANDOM
Number of processes: 2
Process 0       - Ciphertext: SSBAOATJJPOLTGLS  - Decoded(original) text: CASSANDRAHAYDOCK
Process 0       - Ciphertext: DZFMEOCLEL        - Decoded(original) text: AMBERWLEBY
Process 0       - Ciphertext: ZCHXUGKFEQKR      - Decoded(original) text: MATTHEWBROWN
Process 0       - Ciphertext: AYYOAAXVROILDEYRU         - Decoded(original) text: ALEXANDERBOUDREAU
Process 0       - Ciphertext: WVNTXAZXICLRGEJN  - Decoded(original) text: WILLIAMVANLEEWUN
Process 0       - Ciphertext: VEWPSKYKYD        - Decoded(original) text: TREVORHILL
Process 0       - Ciphertext: VKOINCGIUXGHC     - Decoded(original) text: SAMUELBARNETT
Process 0       - Ciphertext: EXQNRWNV  - Decoded(original) text: ALLENKIM
Process 0       - Ciphertext: PAPZWAXUFYBDJOC   - Decoded(original) text: PRISCAONYEBUCHI
Process 0       - Ciphertext: EBZFPJMQFEQYAN    - Decoded(original) text: CAMRYNMOERCHEN
Process 0       - Ciphertext: TTCPTWRUQNNW      - Decoded(original) text: THOMASRICKUS
Process 0       - Ciphertext: OVYZCCQFJIZFP     - Decoded(original) text: MARIAHJOHNSON
Process 0       - Ciphertext: MSJRLSOCLBGI      - Decoded(original) text: KEIARACLARKE
Process 0       - Ciphertext: NRAHKCNR  - Decoded(original) text: SEANROSE
Process 0       - Ciphertext: IELHBNUSHJP       - Decoded(original) text: VICTORLEUNG
Process 0       - Ciphertext: LWHZZXFIEVDSWUVZRGLVMRXFKBEMAAGBRZRSROTVGAWEZVTZGY        - Decoded(original) text: LOEMIPSUMIDKTHERESTIMJUSTTRYINGTOMAKEABIGSTRINGLOL
Process 0       - Ciphertext: UOZLLDRWQAPQNMLOB         - Decoded(original) text: HOWDYDOODAMIAMIGO
Process 0       - Ciphertext: VJSJLRUBCKSVTVCGGNWSOWVECHGXAACDHNDTAVZQXQHQAAZARFIXBKYLYLBKELPVBQUMBJ    - Decoded(original) text: INEEDANOTHERBIGSTRINGFORTESTINGPURPOSESDONTMINDMEJUSTTRYPINGMYTHOUGHTS
Process 0       - Ciphertext: ZXLYRXWOJUVRICUBFMLS      - Decoded(original) text: DOIKNOWANYSONGLYRICS
Process 0       - Ciphertext: UYSSQFUBPEUPUSELTZQEQNDROBCJUCCQUECADLAXMUUIQTOXOXGPLEAYBIHCLHPQFLGE      - Decoded(original) text: THEONLYSONGLRYICSICANTHINKOFRIGHTNOWAREOLDGENZSONGSLIKEPARTYINTHEUSA
Process 0       - Ciphertext: AAQKXLFEUQEVIFQYLHRVTXCMQEWKUDFKWYAAQVPVNMPCIZKYENKFYLCERK        - Decoded(original) text: ANDIPUTMYHANDSUPTHEIRPLAYINGMYSONGANDTHEBUTTERFLIESFLYAWAY
Process 0       - Ciphertext: KQPNTWDOKRPJANUUPHBCTWZEFPSWJQFREVTTVGMR  - Decoded(original) text: NODDINGMYHEADLIKEYEAHMOVINGMYHIPSLIKYEAH
Process 0       - Ciphertext: KRVSIWVVBQSMSXOVWVAJEWRRQMXNUVXHMSZQNPZKFGVJMHPSFVCEQMX   - Decoded(original) text: HEREARESOMEENGLISHSENTENCESWRITTENINALLCAPSWITHNOSPACES
Process 0       - Ciphertext: TZTPVURIVZIFBXPJQRNGRPVGAONWMMMCVHSDJOVHZPUIOSU   - Decoded(original) text: HELLOHOWAREYOUDOINGTODAYWHATAREYOUPROGRAMMINGON
Process 0       - Ciphertext: SPEHJNDMASRNDMVZZEABSAFLNWA       - Decoded(original) text: WHATSTHEWEATHERLIKETOMORROW
Average time taken to encrypt and decrypt all the strings: 814 nanoseconds.
```

We can see from the output that the parallelized version is faster than the non-parallelized version. We can even see from how scaling the number of entries to encrypt/decrypt from 22 to 50, the performance of the parallelized version scales much better. The non-parallelized version's time almost doubles, whereas the parallelized version increased only slightly, around 200 nanoseconds.

The observed performance speedup and scalability of the parallelized version of the Vigenère cipher highlight the significant advantages that High-Performance Computing (HPC) offers to cryptology on a larger scale. As evidenced by the faster execution time and better scaling with increased workload, parallelization enables cryptographers to efficiently handle larger volumes of data and complex cryptographic operations. In the context of HPC, which encompasses clusters of interconnected high-performance computing nodes with powerful processing capabilities, the parallelized encryption and decryption processes can be further optimized and scaled. By leveraging the parallel processing capabilities of HPC systems, cryptologists can tackle computationally intensive cryptographic tasks with unprecedented speed and efficiency. Additionally, the minimal increase in execution time observed when scaling the workload in the parallelized version underscores the robustness and scalability of parallelization techniques, making HPC an indispensable tool for cryptographers seeking to analyze, develop, and deploy sophisticated cryptographic algorithms at scale. Overall, the performance benefits demonstrated by parallelizing the Vigenère cipher underscore the critical role of HPC in advancing cryptology by enabling faster, more scalable, and more secure cryptographic solutions.

References

Arel, R. (2023, May 12). Explore the impact of quantum computing on cryptography: TechTarget. Tech Target. https://www.techtarget.com/searchdatacenter/feature/Explore-the-impact-of-quantum-computing-on-cryptography

Ayub, Md. A., Ahmed Onik, Z., & Smith, S. (2019a). Parallelized RSA algorithm: An analysis with performance evaluation using OpenMP Library in high performance computing environment. 2019 22nd International Conference on Computer and Information Technology (ICCIT). https://doi.org/10.1109/iccit48885.2019.9038275

Bloom, G., & Simha, R. (2012). Cryptographic Accelerator. Science Direct. https://www.sciencedirect.com/topics/computer-science/cryptographic-accelerator

Finke, D. (2024, March 19). Nvidia announces new supercomputer projects, a quantum cloud, academic initiatives, and PQC acceleration at its annual GTC conference. Quantum Computing Report. https://quantumcomputingreport.com/nvidia-announces-new-supercomputer-projects-a-quantum-cloud-academic-initiatives-and-pqc-acceleration-at-its-annual-gtc-conference/

GeeksforGeeks. (2023, May 29). Vigenère cipher. GeeksforGeeks. https://www.geeksforgeeks.org/vigenere-cipher/

Google Cloud Tec. (2021, November 25). How does encryption work at Google's data centers? YouTube. https://youtu.be/82YD6esAK0o?feature=shared

Joux, A. (2012). A Tutorial on High Performance Computing Applied to Cryptanalysis. *Advances in Cryptology – EUROCRYPT 2012*, *7237*. https://link.springer.com/chapter/10.1007/978-3-642-29011-4_1#citeas

Mochurad, L., & Shchur, G. (2021). Parallelization of Cryptographic Algorithm Based on Different Parallel Computing Technologies. https://ceur-ws.org/Vol-2824/paper3.pdf

Security in high-performance computing (HPC): Secure-IC. Secure IC. (2022, March 4). https://www.secure-ic.com/applications/challenges/high-performance-computing/

Timberg, C. (2013, September 6). Google encrypts data amid backlash against NSA spying . The Washington Post. https://www.washingtonpost.com/business/technology/google-encrypts-data-amid-backlash-against-nsa-spying/2013/09/06/9acc3c20-1722-11e3-a2ec-b47e45e6f8ef_story.html

Wolfram U. (2023, November 17). Introduction to cryptography: Substitution-permutation networks. YouTube. https://youtu.be/SBgxbHk5qE8?feature=shared