**Use of Modular Arithmetic in Cryptography**

**Information Theory and Coding**

**IA2 – SY IT A2**

**Hiral Patel –** 16010421071

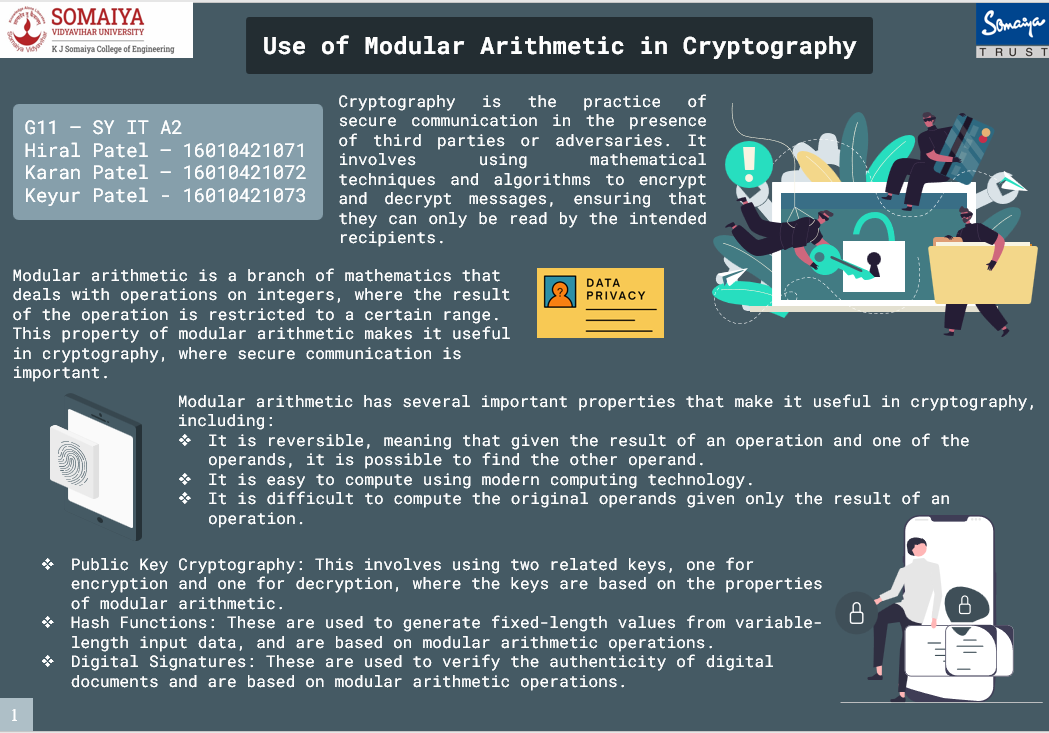
**Karan Patel** – 16010421072

**Keyur Patel** – 16010421073

**Presentation Images:**

**Presentation Explanation for page 1:**

Images:



Content:

**What is Modulo Arithmetic?**

Modular arithmetic is a type of arithmetic in which the operands are numbers that are modulo a certain number. In other words, the operands are reduced to a common denominator. For example, in modular arithmetic, 5 modulo 3 is 2, because 5 divided by 3 is 1 with a remainder of 2. This remainder is what is stored in the modular arithmetic system.

**Why is Modular Arithmetic used in Cryptography?**

* The modulo operation is used in cryptography to create secure communications.
* It is used to create a shared secret between two parties that can be used to encrypt and decrypt data.
* It is difficult to solve modular arithmetic problems, but easy to verify the results. This makes it a good choice for cryptographic algorithms.
* Modular arithmetic is a fundamental concept in cryptography that is used

extensively in many cryptographic algorithms.

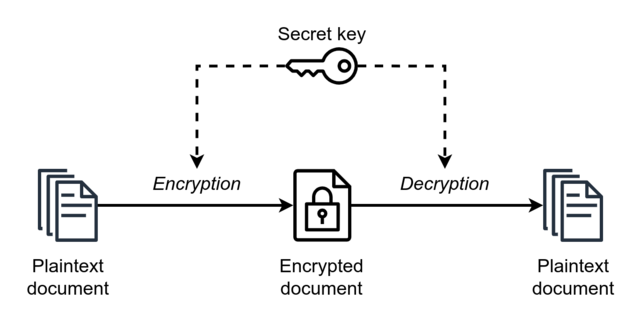
**Here are some of the advantages and disadvantages of using modular arithmetic in Cryptography.**

|  |  |
| --- | --- |
| **Advantages** | **Disadvantages** |
| Modular arithmetic is a well-defined and well-understood mathematical concept that is easy to implement and efficient to  compute. | Modular arithmetic can be vulnerable to attacks such as modular  exponentiation attacks, where an attacker can calculate the private key  from the public key. |
| Modular arithmetic is widely used in cryptography for key  generation, encryption, and decryption algorithms. | Modular arithmetic can be vulnerable to attacks such as small-  subgroup attacks, where an attacker can calculate the private key by  exploiting the structure of the modular arithmetic. |
| Modular arithmetic allows for the creation of one-way functions,  which are essential for creating digital signatures and other  cryptographic protocols. | Modular arithmetic requires the use of large prime numbers, which  can be difficult to generate and manage in some cryptographic  applications. |
| Modular arithmetic provides a high degree of security against  attacks such as brute-force attacks and statistical attacks. | Modular arithmetic can be vulnerable to timing attacks and other  side-channel attacks, where an attacker can observe the timing and  other physical characteristics of the computation to deduce the private  key. |

In summary, modular arithmetic is a powerful and widely used concept in cryptography, but it requires careful implementation and management to ensure that it is secure against known attacks.

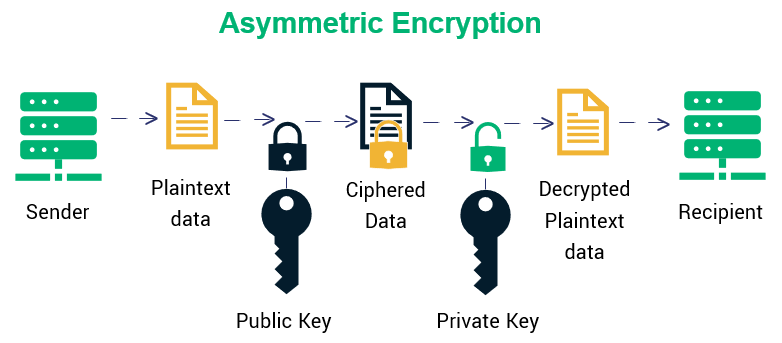
**Types of Cryptography:**

1. **Symmetric Key Cryptography:**



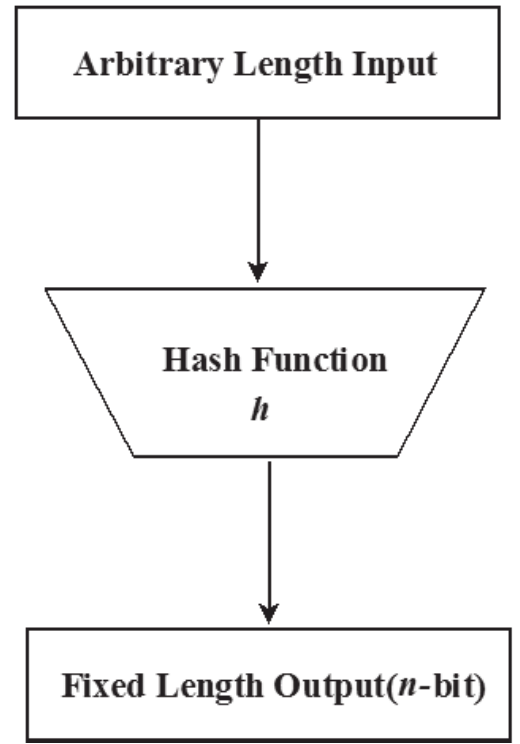
* Symmetric cryptography, also known as secret-key cryptography, uses a single secret key to encrypt and decrypt messages.
* The sender and the receiver both use the same key to encrypt and decrypt messages, making it faster and more efficient than asymmetric cryptography.
* However, the main drawback of symmetric cryptography is that both the sender and the receiver must have access to the same secret key, which can be difficult to manage securely.

1. **Asymmetric Key Cryptography:**



* Asymmetric cryptography, also known as public-key cryptography, uses a pair of keys: a public key and a private key.
* The public key is used to encrypt messages, and the private key is used to decrypt them.
* This allows the sender to encrypt messages without needing to share a secret key with the recipient, making it more secure than symmetric cryptography.
* Asymmetric cryptography is commonly used for digital signatures and key exchange protocols.

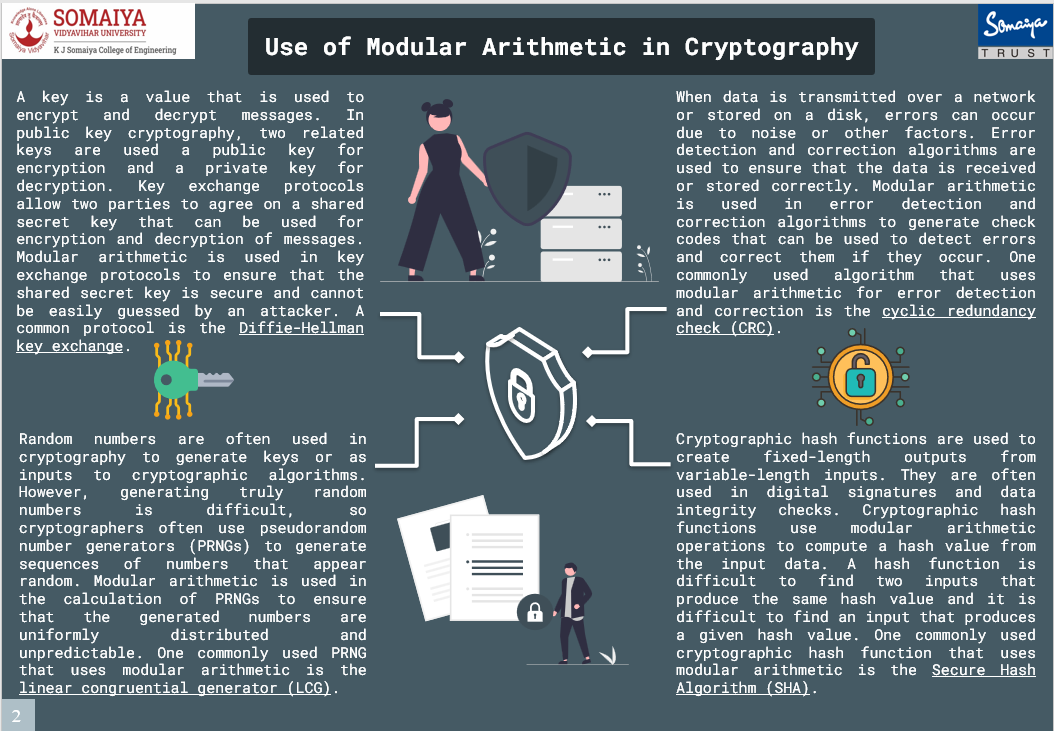
1. **Hash Function Cryptography:**



* There is also a third type of cryptography known as hash functions, which are used to generate fixed-length outputs from arbitrary-length inputs.
* Hash functions are commonly used in digital signatures and to ensure the integrity of data. However, they are not used for encryption or decryption of messages.

**Presentation Explanation for page 2:**

Images:



Content:

**Uses of Modular Arithmetic in Cryptography:**

1. **Public key Cryptography:-**

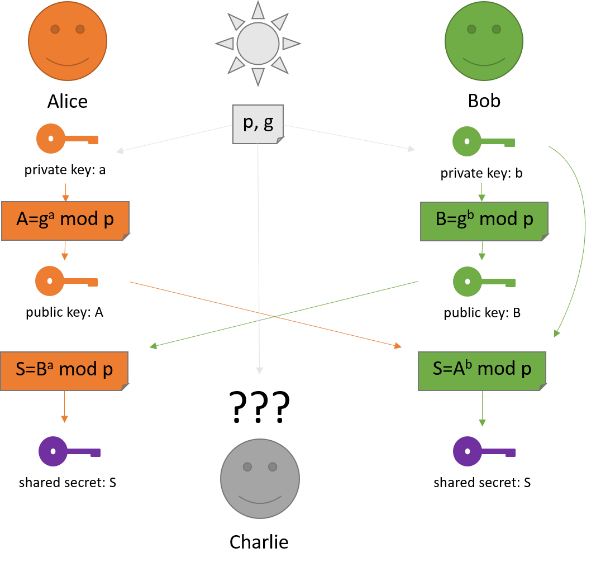
* When someone wants to send a message to the recipient, they use the recipient's public key to encrypt the message.
* Once the message is encrypted, only the recipient's private key can be used to decrypt it.
* This ensures that only the intended recipient can read the message, even if the message is intercepted by an attacker.

Public key cryptography is often used for secure communication over the internet, such as online shopping or online banking. It is also used for digital signatures, which can be used to verify the authenticity of a message or document.

**One such Algorithm used in Public key Cryptography is Diffie-hellman key exchange.**

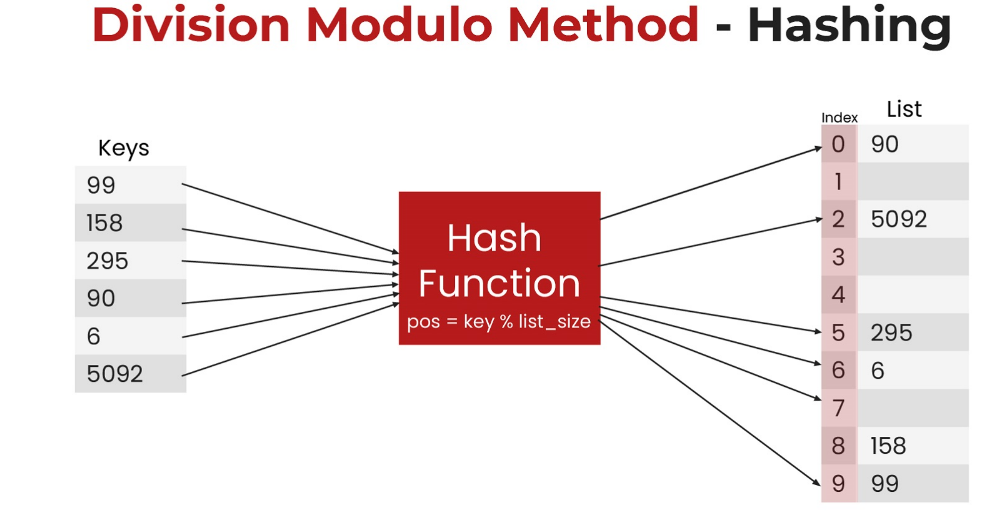
**Diffie-hellman key Exchange**

* The Diffie-Hellman key exchange (also known as exponential key exchange) is a method for securely exchanging cryptographic keys over an insecure channel.
* It works by allowing two parties (Alice and Bob) to agree on a shared secret key without any other party being able to intercept the key or learn anything about it.
* The key exchange involves the following steps −
* Alice and Bob agree on two large prime numbers, **p** and **g**, and a public key exchange algorithm.
* Alice chooses a secret integer, **a**, and computes **A = ga** **mod p**. She sends **A** to Bob.
* Bob chooses a secret integer, **b**, and computes **B = gb** **mod p**. He sends **B** to Alice.
* Alice computes **S = Ba mod p**. Bob computes **S = Ab mod p**.
* Alice and Bob now both have the shared secret key **S**, which they can use to establish a secure communication channel.



The security of the Diffie-Hellman key exchange relies on the fact that it is computationally infeasible for an attacker to determine the shared secret key s from the public values of **p, g, A,** and **B**. This allows Alice and Bob to exchange the key securely, even over an insecure channel.

1. **Hash Function Cryptography:-**



* A hash function is a mathematical function that takes in an input (message or data) and outputs a fixed-size string of characters, known as a hash value or message digest.
* Division Modulo Method is the simplest method of hashing.
* In this method, we divide the element with the size of the hash table and use the remainder as the index of the element in the hash table.

Hash functions have several properties that make them useful for cryptography:

**1. Deterministic:** Given the same input, a hash function will always produce the same output.

**2.** **One-way:** It is computationally infeasible to determine the input message or data from the hash value.

**3.** **Fixed-length output:** The hash function produces a fixed-length output, regardless of the input size.

**4. Collision-resistant:** It is computationally infeasible to find two different inputs that produce the same hash value.

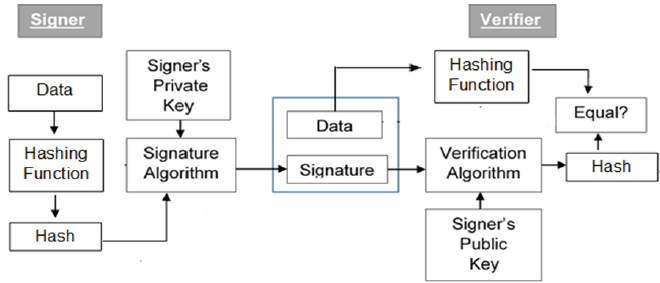
**Applications of Hash function in Cryptography:**

* Hash functions are used in a variety of applications, such as password storage, digital signatures, and message authentication codes (MACs).
* In password storage, a hash function is used to transform a user's password into a fixed-length hash value, which is stored in a database.
* When the user logs in, the system hashes the entered password and compares it to the stored hash value to determine if the password is correct.
* In digital signatures, a hash function is used to generate a message digest of a document, which is then encrypted using the sender's private key. The recipient can use the sender's public key to decrypt the message digest and verify the authenticity of the document.
* In message authentication codes (MACs), a hash function is used to generate a tag or checksum that is attached to a message. The recipient can use the same hash function and key to generate a new tag and compare it to the received tag to verify the integrity of the message.

**What is a Digital Signature?**

* A digital signature is a cryptographic technique used to verify the authenticity and integrity of a digital document, message or transaction.
* It provides a way to sign a document electronically, such that it can be verified that it was signed by a specific person or entity, and that it has not been altered since it was signed.

**Working of Digital Signature using Hash Alogorithm**



1. A digital signature is created by combining a message with the signer's private key, using a mathematical algorithm known as a hash function.
2. This creates a unique digital signature that can be attached to the document.
3. When someone receives the signed document, they can verify its authenticity by using the signer's public key to decrypt the signature and compare it to the original message.
4. If the two match, it is a sign that the document has not been tampered with, and that it was indeed signed by the signer.

**Application of Digital Signature in Real World**

Digital signatures are widely used in e-commerce, financial transactions, and other situations where a signed document or message is required. They offer several advantages over traditional paper signatures, including faster processing, easier verification, and better security.

Some common applications of digital signatures include:

1. Signing contracts and agreements

2. Authenticating emails and other electronic messages

3. Signing software and code to ensure it has not been tampered with

4. Signing legal documents, such as wills and deeds

5. Signing government documents, such as tax returns and permits.

1. **Linear Congruential Generator:-**

* A linear congruential generator (LCG) is a simple algorithm used to generate a sequence of pseudo-random numbers.
* It is not typically used for key exchange as it is considered insecure due to predictable patterns in the generated numbers.
* However, for educational purposes, here's an example of how LCG key exchange could work:

1. Alice and Bob agree on a modulus **m**, a multiplier **a**, an increment **c**, and a seed value **X0**. These values are known to both of them and can be shared over an insecure channel.

2. Alice chooses a secret random value kA and calculates **KA** = (a \* kA + c) mod m. She sends **KA** to Bob over the insecure channel.

3.Bob chooses a secret random value kB and calculates **KB** = (a \* kB + c) mod m. He sends **KB** to Alice over the insecure channel.

4. Alice calculates the shared secret key **S** = (KB \* kA + c) mod m.

5. Bob calculates the shared secret key **S** = (KA \* kB + c) mod m.

* Now Alice and Bob have a shared secret key S that can be used to encrypt and decrypt messages.
* However, the security of this key exchange method is compromised by the fact that LCGs can exhibit predictable patterns in their output.
* An attacker who can predict the output of the LCG could determine the shared secret key and intercept messages.
* Therefore, LCGs are not considered suitable for key exchange in practice and more secure algorithms, such as Diffie-Hellman or Elliptic Curve Cryptography, are used instead.

1. **Cyclic Redundancy Check(CRCs) :-**

Cyclic Redundancy Check (CRC) is an error detection technique used in digital communication. It is used to verify the integrity of data transmitted over a network or stored in memory.

Here is an example of how CRC works:

Suppose we want to transmit the message "**110101**" over a communication channel. To detect any errors in transmission, we can use CRC.

The CRC algorithm uses a divisor polynomial to generate a checksum, which is appended to the message.

1. Choose a generator polynomial. The generator polynomial is a fixed value that is agreed upon by both the sender and receiver. It is usually a binary number represented as a polynomial. For example, the generator polynomial could be **1011**, which represents **x3 + x + 1.**

2. Append zeros to the message. The number of zeros to be appended is equal to the degree of the generator polynomial. In our example, we need to append three zeros to the message "**110101**" to get "**110101000**".

3. Divide the message by the generator polynomial using modulo-2 division. This generates a remainder that is used as the checksum. In our example, the message "**110101000**" divided by the generator polynomial "**1011**" gives a remainder of "**111**".

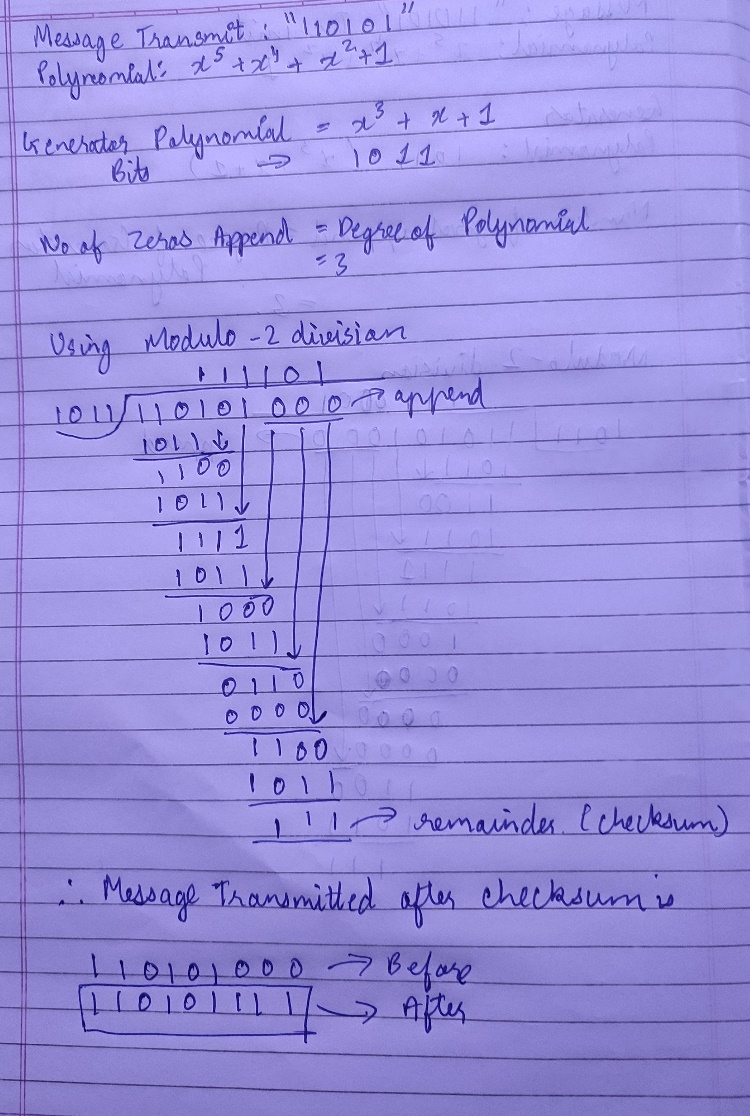
4. Append the checksum to the original message. In our example, the message with the checksum is "**110101111**".

5. Transmit the message with the checksum over the communication channel.

6. At the receiving end, the receiver performs the same division using the same generator polynomial.

If the remainder obtained is zero, it means that there were no errors in transmission.

If the remainder is nonzero, it indicates that there was an error in transmission and the message needs to be retransmitted.



**Applications of CRCs**

* CRC is a widely used error detection technique and is used in many communication protocols such as Ethernet, Wi-Fi, Bluetooth, and others.
* It is a simple and efficient technique for detecting errors in transmission and ensures the integrity of data

References:

1. <https://www.geeksforgeeks.org/linear-congruence-method-for-generating-pseudo-random-numbers/>
2. <https://www.tutorialspoint.com/cryptography/cryptography_digital_signatures.htm>
3. <https://technicalsand.com/hashing-in-data-structure/>
4. <https://www.techtarget.com/searchsecurity/definition/Diffie-Hellman-key-exchange>
5. <https://en.wikipedia.org/wiki/Diffie%E2%80%93Hellman_key_exchange>
6. <https://www.educba.com/diffie-hellman-key-exchange-algorithm/>
7. <https://www.tutorialspoint.com/what-is-modular-arithmetic-in-information-security#:~:text=Modular%20arithmetic%20enables%20us%20to,different%20groups%20which%20can%20work>.