ANALYSIS NUMERICAL THEORY ANALYSIS FOR CRYPTOGRAPHY AND SECURITY APPLICATIONS

Introduction

Assuring the confidentiality, integrity, and validity of data has grown to be of the utmost importance in the age of digital communication and information exchange. Sensitive information can be protected against unauthorized access or manipulation using a variety of techniques and algorithms provided by cryptography, the science of secure communication. Number theory, a discipline of pure mathematics that studies the characteristics and connections of integers, is at the foundation of many cryptographic systems. Prime numbers, modular arithmetic, and integer characteristics are among the foundational ideas of number theory that have proven to be extremely useful in the development and analysis of cryptographic algorithms. Numerous key exchange protocols, digital signature systems, and encryption and decryption approaches all have their theoretical foundations in number theory.

This essay seeks to give a thorough overview of number theory's uses in security and cryptography. It looks at how number theory ideas can be used to create cryptographic algorithms and protocols that can fend off attacks like brute-force, factorization, and discrete logarithm problems. The implications of computational complexity and new dangers posed by quantum computers are also discussed, along with the limitations and practical considerations of number theory-based encryption systems. Primitive numbers and modular arithmetic are two of the fundamental concepts of number theory that are introduced at the outset of the analysis. The importance of prime numbers in cryptography is discussed, with a focus on how they help create secure cryptographic keys and implement key-exchange protocols that allow for private communication between parties.

1. Related Work

An introduction to the use of number theory in cryptography. Prime numbers, modular arithmetic, the Euclidean method, the Chinese Remainder Theorem, Euler's Theorem, the RSA algorithm, and the discrete logarithm issue are some of the subjects it addresses

This survey paper provides a thorough introduction of public key cryptography with an emphasis on its mathematical underpinnings.

It covers subjects including identity-based cryptography, RSA, Diffie-Hellman, El Gamal, elliptic curve encryption, digital signatures, and key exchange protocols.

Lattice-based cryptography, which offers protection against assaults from both classical and quantum computers, is the subject of this comprehensive paper. It covers the fundamentals of lattices, lattice-based key exchange protocols, lattice-based encryption techniques and lattice-based signature schemes. The goal of post-quantum cryptography, the subject of this comprehensive study, is to create cryptographic algorithms that are impervious to attacks from quantum computers.

It encompasses several families of post-quantum cryptography systems, including isogeny-based, lattice-based, code-based, and hash-based ones. It includes subjects including side-channel attacks, fault attacks, collision attacks, linear cryptanalysis, algebraic cryptanalysis, and differential cryptanalysis on cryptographic hash functions, among others.

2. Fibonacci Sequence in Cryptography

It has been investigated if the Fibonacci sequence, which consists of the numbers 0, 1, 1, 2, 3, 5, 8, and 13, may be used in cryptography. Each number in the sequence is the sum of the two numbers before it. Even while it might not be utilized as frequently as other cryptographic methods, a cryptosystem built on the Fibonacci sequence can offer an intriguing substitute in some situations. Here are some specifics on a fundamental application of a Fibonacci-based cryptosystem [9]:

Generating a Key:

1. The cryptosystem starts by choosing two initial parameters, usually two consecutive Fibonacci numbers, to serve as the secret key. For instance, if n is a suitable large number, we can choose F(n) and F(n+1) as the starting parameters.
2. Making use of the Fibonacci Sequence Start with the initial settings and generate a Fibonacci sequence until it reaches the desired length. The next number can be obtained by repeatedly adding both of the final numbers in the sequence. Both encryption and decryption will take place using the produced sequence.

Encryption:

1. Message Representation: Transform the plaintext message into numerical form using a suitable method, such as mapping to numerical values or ASCII representation.
2. Process of Encryption: Use the Fibonacci sequence to encrypt the message's numerical form. Iterate through the Fibonacci sequence for each numerical value in the message, and for each Fibonacci number you come across, add it to the original value.
3. Ciphertext generation: The ciphertext is made up of the resulting changed numerical values.

C = P * F(i) (1) Decryption:

1. Retrieve the numerical ciphertext values after processing the ciphertext.
2. The Fibonacci sequence is used to decrypt the numerical ciphertext. Continue iterating through the Fibonacci sequence, taking each Fibonacci number out of the numerical value as you go.
3. Get the resulting numerical numbers and transform them back into their original plaintext form to get the message.

The Fibonacci sequence, a well-known series that begins with 0 and 1, is created by adding the two terms before it. A revised version of the Fibonacci sequence is provided below:

$$\{0, 1, 1, 2, 3, 5, 8, 13, \ldots\}$$

Each term in this series is the product of the two terms before it. For instance, adding 1 and 1 yields 2, adding 1 and 2 yields 3, adding 2 and 3 yields 5, and so on. Each term in the sequence is the product of the two terms before it, and the sequence never ends.

The following is a definition of the Fibonacci sequence:

For all non-negative integers n, F(0) = 0, F(1) = 1, F(n+2) = F(n) + F(n+1), and so on.

The series begins with F(0) = 0 and F(1) = 1 in this definition. The two terms F(n) and F(n+1) that come before it is added together to create the next term, F(n+2). The

Fibonacci sequence is produced by this recursive formula, where each term is the sum of the two terms before it.

Property: $F(n+2)= F(0)+ F(1)+ F(2)+ \ldots + F(n)+ F(n+1)$

The Golden ratio (denoted by the Greek symbol phi) served as the foundation for Binet's explicit formula for the Fibonacci sequence, which was developed in 1843. The following is an expression for the formula:

$$F(n)=((\varphi^n- (1-\varphi)^n))/\sqrt{5}$$

Proposition:

Let α and β be the roots of the equation $x2 = x + 1$, where and are determined as follows:

$$\alpha = (1 + \sqrt{5}) / 2$$

$$\beta = (1 - \sqrt{5}) / 2$$

The Fibonacci numbers
can therefore be described
using and as follows:

$$F(n) = (\alpha^n - \beta^n) / \sqrt{5}$$

In this illustration, the nth Fibonacci number is represented by $F(n)$, and the roots of the equation $x2 = x + 1$ are and. By raising and to the power of n, deducting the results, and dividing by the square root of 5, the formula is completed. It is feasible to immediately calculate the value of any Fibonacci number using this method without the need for iterative or recursive calculations.

Leema: The Fibonacci sequence is indeed not a super increasing sequence. Specifically, any consecutive finite subsequence $\{Fm, Fm+1, ..., Fm+r\}$ where $r > 2$ and $m > 0$ is not a super increasing sequence.

Proof: It is possible to show that the Fibonacci sequence is rather than a super increasing sequence by using a counterexample. Let's think about the sequence's first few terms: 0, 1, 1, 2, 3, 5,... It is evident from observation that 3 is less than the total of the preceding terms, which is equal to $0 + 1 + 1 + 2$. Alternatively put, $F4 = F0 + F1 + F2 + F3$.

Examining the subsequence "Fm, Fm+1,..., Fm+r" for $m > 0$ and $r > 2$ is the next step. To show that the subsequence's term sum does not exceed the next Fibonacci number, we can do a quick check. Particularly, we have

$$Fm + Fm+1 + ...+ Fm+r < Fm+2$$

It can be further simplified as

$$Fm+r < Fm+2 - (Fm + Fm+1)$$

Since the Fibonacci sequence is recursive, $Fm+2 - (Fm + Fm+1)$ can be represented as $Fm+1$. As a result, we have:

$$Fm+r < Fm+1$$

This counterexample demonstrates that the sum of each term in the subsequence "Fm, Fm+1,..., Fm+r" is not larger than the following Fibonacci number, $Fm+1$, for any $m > 0$ and $r > 2$. The sequence of Fibonacci numbers does not meet the super increasing property as a result.

The Fibonacci sequence does not fit the criteria of a super increasing sequence, despite the fact that it has unique qualities and traits of its own.

The Fibonacci sequence is comparable to the Lucas sequence, represented as Ln0, although the Lucas series begins with the numbers 2 and 1. After the first two numbers, it recurs according to the same rules as the Fibonacci sequence. L0 = 2 and L1 = 1 by definition, and for n > 0, Ln+2 = Ln + Ln+1.

The Lucas sequence can therefore be illustrated as follows:

$$\{2, 1, 3, 4, 7, 11, ...\}$$

Remember that the Lucas sequence is not a super increasing sequence, just like the Fibonacci sequence is not. A super increasing sequence, as was previously mentioned, demands that each term be greater than the total of all previous terms. The Lucas sequence, however, does not meet this requirement.

$$Ln = F(n-1) + F(n+1)$$

The nth term of the Lucas sequence is represented by Ln, and the nth term of the Fibonacci sequence is represented by F(n). Using this method, it is possible to compute the Lucas numbers based on the corresponding Fibonacci numbers and to directly connect the Lucas series to the Fibonacci sequence.

3. Cryptosystem Methodology

Based on the Knapsack Cryptosystem and the idea of super increasing sequences, we have created two cryptosystems:

Using a pre-shared key cryptosystem

- Before any communication or interaction can occur in this cryptosystem, Alice and Bob must both have a pre-shared key in the form of a super increasing sequence.
- The common key for encryption and decoding is a super increasing sequence.
- This key can be used by Alice and Bob to securely encrypt and decrypt messages sent back and forth.

Cryptosystem for exchanging keys:

- With this cryptography, Alice and Bob can create a shared key while still in communication.
- They are not need to have a pre-shared key at first.
- In a key exchange protocol, Alice and Bob trade information in order to generate a single super increasing sequence that serves as their shared key.

During their communication, this shared key might be used for later encryption and decryption procedures.

Secret Key provide:

The super increasing sequence vector r = (n, 2n,..., rn) in the provided cryptosystem scenario represents the shared key that Alice and Bob share. The values g Z+ and p, a huge prime number satisfying p > 2rn, are included in the public parameters. gcd(a, p - 1) = 1 for Alice's secret key a Z+, and gcd(k, p - 1) = 1 for Bob's secret key k Z+.

In this cryptosystem, the encryption and decryption procedures are as follows:

1. Alice calculates A = ga mod p and sends Bob the result, A.
2. Bob wants to encrypt a plaintext message with the following format: x Zq (where q is an appropriate modulus for the plaintext space). He creates two parts of the ciphertext:

   C1 = p mod gk

   C2 = Ak(x r) mod p,

   where stands for the action of doing a dot product between two vectors.
3. Bob gives the ciphertext parts (C1, C2) to Alice, who then begins the decryption process.

Using her secret key a, she calculates C' = (C1)-a C2 mod p. The original plaintext message x is then retrieved by Alice by employing the super increasing sequence r and the value gained from it to solve the Knapsack Problem.

Proof:

$$(\mathrm{ci})^{(-\mathrm{a})(\mathrm{c2})\mathrm{mod}}\mathrm{p} = \left(\mathrm{g}^{\mathrm{ka}}\right)^{\mathrm{a}^{-1}} * \left(\mathrm{g}^{\mathrm{akx}\cdot\mathrm{r}}\right)\mathrm{mod}\ \mathrm{p} = \mathrm{g}^{\mathrm{kx}\cdot\mathrm{r}}\mathrm{mod}\ \mathrm{p} = \mathrm{x}\cdot\mathrm{f}\ \mathrm{mod}\ \mathrm{p}$$

Since p > 2rn, the equation xf mod p can be converted into a Knapsack problem using the vector (x • r) and the super increasing sequence r. This gives us the opportunity to use Proposition 2.2, which can be used to find x.

Example:

We may rebuild the procedure as follows based on the information provided and the stages described in the scenario:

A is sent to Bob once Alice has calculated it as A = ga mod p = 997 mod 1223 = 856.

Bob holds the message x = (0, 0, 1, 1, 0, 0, 1, 0) in plaintext. According to his calculation of (C1, C2),

C1 = g^k mod p = 99^3 mod 1223 = 460

C2 = A^(x* f) mod p = 856^(x* f) mod 1223 = 45 Alice receives
(460, 45) from Bob.

Calculated by Alice, C' = (C1)(-a) * 45 mod 1223 = 233 * C2 mod p: C' = 460(-7) *

In order to retrieve the plaintext message x = (0, 0, 1, 1, 0, 0, 1, 0), Alice then solves the Knapsack Problem for (r, 233).

As a result, Alice decrypts the ciphertext successfully and obtains the original plaintext.

The above cryptosystem is based on the fact that Alice and Bob already have a common super increasing sequence. This is a fairly large assumption so we have created another similar cryptosystem that creates a common super increasing sequence based on Fibonacci subsequence.

4. Conclusion

Number theory is a key component in the development and evaluation of cryptographic systems. Numerous cryptographic techniques and protocols are built on the study of prime numbers, modular arithmetic, and other number-theoretic ideas. We have examined the usefulness of number theory in applications for cryptography and security throughout this examination. We talked about the super increasing sequences-based Knapsack Cryptosystem's restrictions in terms of attack weaknesses. We also looked at the Lucas sequence and how it relates to the Fibonacci sequence, emphasizing its use in cryptography techniques. These cryptosystems have a level of security due to the employment of super increasing sequences in the key generation and encryption procedures. Number theory provides useful methods and instruments for creating safe cryptography systems. However, in an increasingly connected and digital world, it is crucial to keep up with the most recent advancements in the industry and employ more reliable encryption methods to preserve the security and integrity of critical information.