International Journal of Multidisciplinary Innovative Research
ISSN: 2583-0228 Volume 4, Number 1 (Jan' 2024) pp. 8-14
© CIIR, Noida, INDIA (www.ciir.in)
https://www.ijmir.org
Article Received: 5 Oct' 2023; Accepted: 6 Nov' 2023.

Review Article

RSA Encryption: Bridging Ancient Mathematics with Modern Cybersecurity

T. N. Kavitha

Assistant Professor, Department of Mathematics, Sri Chandrasekharendra Saraswathi Viswa Mahavidyalaya, Enathur, Kanchipuram, Tamilnadu, India.

Email: tnkmaths@gmail.com

ABSTRACT: ***Virtually all facets of contemporary life have changed as a result of the quick development of digital technology, which has had an impact on a variety of industries including communication, business, science, and entertainment. This study demonstrates how ancient ideas such as number systems, geometry, algebra, and algorithms serve as the foundation for contemporary notions such as data compression, artificial intelligence, computer graphics, and more. It becomes clear that ancient civilizations' mathematical prowess served as an important link between the past and the present, providing insights that not only deepen our understanding of the past but also equip us to successfully traverse the intricacies of today's digital landscape. This work also emphasizes the mutually beneficial interaction between traditional mathematics and modern technologies. As digital trends develop, they frequently provide fresh viewpoints on age-old mathematical conundrums, igniting new research into subjects that have captivated mathematicians for decades. On the other hand, the timeless aspect of mathematical thought is highlighted by the fact that novel solutions to contemporary computer problems continue to be inspired by old mathematical ideas. The interaction between classical mathematics and contemporary digital trends exemplifies the lasting significance of fundamental mathematical ideas in determining the course of technological advancement were discussed. Recognizing and utilizing this historical knowledge can result in more reliable, effective, and morally sound digital systems, providing a comprehensive strategy that combines the past knowledge with the present-day innovations.***

KEYWORDS: ***Contemporary life, Digital technology, Classical mathematical principles, Digital revolution, Historical mathematical inventions.***

INTRODUCTION

Numerous mathematical ideas and techniques with roots in classical mathematical principles have supported this digital revolution. This essay investigates how ancient mathematics continues to inform and propel contemporary digital trends. We explore historical mathematical inventions from Babylonian, Egyptian, Greek, and Indian cultures in order to elucidate the fundamental concepts that still hold true in modern digital applications [1].

Finally, the interaction between classical mathematics and contemporary digital trends illustrates the lasting significance of fundamental mathematical ideas in determining the course of technological advancement. A comprehensive strategy that combines the knowledge of the past with the creativity of the present is possible by acknowledging and utilising this historical expertise to create digital systems that are more reliable, effective, and morally sound [2]. A fascinating story that demonstrates the everlasting nature of mathematical mind is the subtle

interplay between the mathematical creativity of cultures like Babylonian, Egyptian, Greek, and Indian and the constantly changing digital trends of today. The genealogy of these concepts spans centuries and continues to pulsate with life in the algorithms, encryption techniques, artificial intelligence systems, and countless other digital phenomena that form our modern world. This is true of both the clever number systems of the past and the complicated algorithms of the present [3].

This paper sets out on a quest to untangle the complex webs connecting classical mathematics and contemporary digital trends. We want to shed light on how these seemingly archaic notions have found surprising application in the most cutting-edge technologies of today by exploring the historical roots of mathematical concepts. Additionally, this investigation will throw light on how current problems are shedding fresh light on past mathematical concerns in addition to illuminating the contributions of ancient mathematics to modern digital trends. It becomes clear as we move through this meeting of the past and present that the applicability of ancient mathematics to the world of digital trends is not merely a coincidence but rather a harmonious symphony of human intellectual effort across time.

METHODOLOGY

The following examples show how classical mathematics is still applicable to modern digital trends:

Number Systems and Cryptography:

Ancient number systems like the Babylonian base-60 system and the Roman numerals served as the basis for the development of the binary and hexadecimal systems used in modern computing. The foundation for encoding and transferring digital data is provided by these systems [4]. Furthermore, modern cryptographic algorithms that safeguard digital transactions and communication use ideas like modular arithmetic, invented by ancient mathematicians.

Geometry and Computer Graphics:

Mathematicians like Euclid established the foundation for geometric principles that are still essential to computer graphics and visualization today in ancient Greek geometry. Ancient geometric ideas serve as the foundation for modern algorithms for visualizing three-dimensional objects, calculating lighting effects, and mimicking lifelike movements.

Algorithms and Algorithmic Thinking:

The algorithmic logic utilised in modern computing is similar to the algorithmic thinking used by ancient mathematicians to solve problems, such as the Eratosthenes sieve for locating prime numbers [5]. Computer programmes are built around algorithms, which are responsible for everything from data searching and sorting to machine learning and artificial intelligence.

Trigonometry and Digital Signal Processing:

The study of correlations between triangle's angles and sides led to the development of trigonometric functions, which are now essential to digital signal processing. Applications include filtering noise and improving image quality, as well as encoding audio and video data.

Statistics and Data Analysis:

Data analysis, machine learning, and artificial intelligence are the current counterparts to the statistical analysis notion, which has its roots in early records of data collection in ancient

societies. Many different fields still use mathematical methods like regression, probability distributions, and hypothesis testing to aid in decision-making.

Optimization and Operations Research:

Ancient optimization issues, like the shortest path conundrum that the Greeks addressed, have developed into sophisticated operations research. These principles are still in use today, driving the optimization of resource allocation, transportation networks, and supply chains.

Ancient Algorithms and Modern AI:

The effectiveness of contemporary machine learning algorithms is similar to the ancient Egyptian technique for multiplication by doubling and halving. These ancient methodologies for process optimization are compatible with modern AI methods.

Ethical Considerations:

Modern debates regarding the ethics of algorithms, bias in AI, and the equitable use of technology have echoes of the ethical conversations and considerations surrounding mathematical principles in antiquity, such as fairness in taxes or distribution. The relationship between prehistoric mathematical concepts and contemporary digital trends is clear in each of these cases [6]. These examples demonstrate the timeless value of the fundamental mathematical ideas that form the basis of our modern digital world.

The following examples in each of the categories show how modern digital trends and classical mathematics are related:

1. Number Systems and Cryptography:

Use a public key to encrypt a message using the RSA encryption technique, which is based on the mathematical ideas of prime factorization, and a private key to decrypt it. A popular technique for secure communication over an unsecure channel is the RSA encryption algorithm. It is based on huge prime numbers' mathematical characteristics. Here is a simplified, step-by-step procedure for utilizing the RSA method to encrypt a message and the accompanying private key to decrypt it.

Step 1: Key Generation

1. Choose two distinct prime numbers, let's say p = 61 and q = 53.
2. Compute $n = p \times q = 61 \times 53 = 3233$.
3. Calculate Euler's totient function $\phi(n)=(p-1) \times (q-1) = 60 \times 52 = 3120$.
4. Choose an integer e and $(1<e<\phi(n))$ such that e and $\phi(n)$ are coprime. A common choice is e = 17.
5. Calculate the modular multiplicative inverse d of e modulo $\phi(n)$ such that $d \times e \equiv 1 \bmod \phi(n)$. In this case, d = 2753.

Public Key: (e,n)=(17,3233)

Private Key: d=2753

Step 2: Encryption

Let's say you want to encrypt the message "HELLO".

a. Represent each character in the message as a number using a mapping (e.g., A=1, B=2, ..., Z=26). So, "HELLO" becomes [8,5,12,12,15].

b. To encrypt each number m, calculate $c = m^e \mod n$ using the recipient's public key.

For "HELLO":

- $c_1 = 8^{17} \mod 3233 = 2578$
- $c_2 = 5^{17} \mod 3233 = 2714$
- $c_3 = 12^{17} \mod 3233 = 2081$
- $c_4 = 12^{17} \mod 3233 = 2081$
- $c_5 = 15^{17} \mod 3233 = 1879$

The encrypted message is [2578,2714,2081,2081,1879].

Step 3: Decryption

To decrypt each encrypted number c, calculate $m = c^d \mod n$ using the recipient's private key. For each c, calculate $m = c^d \mod n$, and then map m back to the corresponding letter using the inverse mapping.

Note: The actual application of RSA entails extra processes, padding techniques, and security considerations. For the sake of simplicity, the numbers in this example are also quite small. In order to maintain security, RSA really employs other methods and even greater prime numbers. The asymmetric cryptographic method known as RSA, or Rivest-Shamir-Adleman, was developed by Ron Rivest, Adi Shamir, and Leonard Adleman. Particularly for safe communication across unsecure channels like the internet, it is one of the most well-known and secure ways for encrypting and decrypting data.

A public key and a private key are used in asymmetric cryptography, sometimes referred to as public-key cryptography. Although these keys are mathematically connected, it is computationally impossible to determine which one is which. The security of RSA is dependent on how tough it is to factor huge composite numbers into their prime factors as the numbers get larger.

Here's how the RSA algorithm works at a high level:

Key Generation:

Two distinct prime numbers, p and q, are chosen.

- The product $n = p \times q$; is calculated. n is used as part of both the public and private keys.
- The Euler's totient function $\phi(n) = (p-1) \times (q-1)$ is calculated.
- A public exponent e is chosen, typically a small prime such as 3 or 17, such that $1<e<\phi(n)$ and e is coprime with $\phi(n)$.
- The private exponent d is calculated such that $d \times e \equiv 1 \mod \phi(n)$. This can be done using the extended Euclidean algorithm.

Encryption:

- The sender uses the recipient's public key (e, n) to encrypt the message.
- The message is split into blocks, and each block is treated as a number.
- Each block m is encrypted using the formula $c \equiv m^e \mod n$.
- The encrypted blocks c is sent to the recipient.

Decryption:

- The recipient uses their private key d to decrypt the encrypted message.

- Each encrypted block c is decrypted using the formula $m \equiv c^d \mod n$.
- The decrypted blocks are combined to reconstruct the original message.

Secure communication, digital signatures, and authentication are just a few uses for RSA. Because it is challenging to factor huge semiprime numbers into their prime factors, it is secure. Key lengths must be adjusted to retain the same level of security as computers become more powerful. Although RSA is regarded as secure when used with lengthy keys, future developments in quantum computing may have an impact on its security.

2. Geometry and Computer Graphics:

Create a programme that manipulates a 2D image using geometric transformations (translation, rotation, and scaling) and outputs the results on a computer screen. Below is a Python-based method for applying the matplotlib library's geometric transformations (translation, rotation, and scaling) on a 2D image and showing the results.

Python Code:

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.image import imread
import matplotlib.transforms as transforms

# Load the original image

image_path = 'path_to_your_image.jpg'  # Replace with the path to your image file
original_image = imread(image_path)

# Create a figure to display the original and transformed images fig, axes = plt.subplots(1, 3, figsize=(12, 4))

# Display the original image

axes[0].imshow(original_image)
axes[0].set_title("Original Image")

# Translation transformation

translation = transforms.Affine2D().translate(50, 30)
translated_image = translation.transform_affine(original_image)
axes[1].imshow(translated_image)
axes[1].set_title("Translated Image")

# Rotation transformation

rotation = transforms.Affine2D().rotate_deg(30)
rotated_image = rotation.transform_affine(original_image)
axes[2].imshow(rotated_image)
axes[2].set_title("Rotated Image")

# Adjust layout and display the images

plt.tight_layout()
```

```
plt.show()
```

Replace "path_to_your_image.jpg" in this solution with the actual path to your 2D image file. Imread is used to load the image, Affine2D is used to perform translation and rotation changes, and Matplotlib is used to display the original, translated, and rotated images side by side.

3. *Algorithms and Algorithmic Thinking*:

Create a programme that quickly sort a huge list of numbers using the quicksort algorithm. Here's a Python implementation of the quicksort algorithm to efficiently sort a list of numbers:

Python code:

```
def quicksort(arr):
    if len(arr) <= 1:
        return arr

    pivot = arr[len(arr) // 2] # Choose a pivot element
    left = [x for x in arr if x < pivot]
    middle = [x for x in arr if x == pivot]
    right = [x for x in arr if x > pivot]

    return quicksort(left) + middle + quicksort(right)

# Example usage

input_list = [3, 6, 8, 10, 1, 2, 1]
sorted_list = quicksort(input_list)
print("Sorted List:", sorted_list)
```

The 'quicksort' function in this example recursively breaks the input list into smaller sublists based on a pivot element. Elements are divided into three sublists: those less than the pivot, those equal to the pivot, and those more than the pivot. The pivot element is chosen as the centre element in the list. The sublists are joined to create the sorted list after the process continues until the base case is achieved (when the list contains one or zero elements).

While quicksort is often a fairly effective sorting algorithm, in cases where the pivot selection is not ideal, its worst-case time complexity can be $O(n^2)$. Performance can be increased by employing optimizations such picking a solid pivot and utilising different sorting algorithms like mergesort for smaller sublists.

4. *Trigonometry and Digital Signal Processing*:
Create a digital filter using Fourier analysis and frequency domain modification to eliminate noise from an audio recording.

5. *Statistics and Data Analysis*:
Utilize statistical techniques to forecast future purchasing trends by analysing a dataset of consumer purchases to find patterns and correlations.

6. *Optimization and Operations Research*:
Use a genetic algorithm to identify the shortest path between a starting city and all of the destinations in the travelling salesman issue.

7. *Ancient Algorithms and Modern AI*:
Optimize the parameters of a neural network in a machine learning model by using the idea of gradient descent, which was influenced by the ancient Egyptian technique of approximation.

8. Ethical Considerations:
Analyzing a dataset for indications of unequal treatment can let us explore the effects of bias in algorithms and suggest changes to promote fair decision-making.

These sample issues show how classical mathematical concepts might be used to address current issues in digital technology, including cryptography, data analysis, optimization, and artificial intelligence. The solutions to these issues demonstrate the old mathematical ideas' continued applicability in the current digital environment.

CONCLUSION

In conclusion, the investigation of many computational issues and the solutions thereto demonstrated the intricate connection between classical mathematical ideas and modern digital developments. It is clear that the importance of ancient mathematics continues to be ingrained in the very core of current technology, as seen by the clever encryption algorithms of ancient civilizations and the effective sorting strategies developed through algorithmic discoveries. Ancient number systems, geometric ideas, and algebraic techniques have been adopted in digital cryptography, computer graphics, and data analysis with ease. The algorithmic approach that mathematicians used in the past laid the path for the creation of complex AI systems and optimization techniques that underpin modern technological developments.

REFERENCES

[1] William Stallings, "Cryptography and Network Security: Principles and Practice".

[2] Jonathan Katz and Yehuda Lindell, "Introduction to Modern Cryptography: Principles and Protocols".

[3] Simon Singh"The Code Book: The Science of Secrecy from Ancient Egypt to Quantum Cryptography".

[4] Steven Strogatz, "The Joy of x: A Guided Tour of Math, from One to Infinity".

[5] John Stillwell, "Mathematics and Its History".

[6] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein "Introduction to Algorithms".