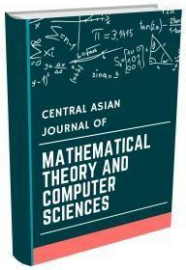

CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES
https://cajmtcs.centralasianstudies.org/index.php/CAJMTCS
Volume: 05 Issue: 03 | July 2024 ISSN: 2660-5309

Article

Study on Computer Applications of Number Theory in Cryptography

Rahul Das, Ashok Kumar Mahato, Suresh Kumar Sahani*

Department of Science, Technology and Engineering, Rajarshi Janak University Janakpurdham, Nepal
* Correspondence: sureshsahani@rju.edu.np

Abstract: Technology has transformed the way we access and store information, making data retrieval fast and efficient. However, safeguarding this information is critical to prevent unauthorized access. Cryptography, rooted in number theory, plays a vital role in data protection by enabling encryption and decryption processes. This study aims to explore various cryptographic algorithms, examining their mechanisms and effectiveness in securing data. Through a comprehensive review of literature from books, journals, and other sources, this research will provide insights into the strengths and weaknesses of different cryptographic methods. The findings will contribute to enhancing data security practices in the digital age.

Keywords: Cryptography, Data Security, Encryption, Decryption, Cryptographic Algorithms

1. Introduction

Information plays an important role in our lives It helps us make choices. We can get information in different ways — like trying out ourselves, talking friends, or reading letters & news articles. Just remember, when you share personal opinions, it's good to say they are opinions! Today's tech world makes it way easier to find information. You can grab it quickly from the internet, watch videos, or check other sources. Also, tech lets us save a lot of stuff that's hard to remember or really important. But we have to be careful. Keeping private info safe is super important so no one else can sneak a peek at it. Tech developers work hard to find better ways to protect our information [1].

They look for solutions until they discover something that can help process or change data; this is known as cryptography! It uses something called Number Theory — a bit of math magic! An exciting paper is in the works titled "Application of Number Theory in Cryptography for Data Security." This paper will show how number theory can help keep our data safe in fun ways. Math is essential in life, and number theory is a big part of it! This area of math is useful all over the place in everyday life. Even though technology moves fast and has its perks, we should also keep an eye on its downsides [2].

Tech usually aims to make things easier & quicker for us. But if someone uses tech for bad reasons, it could lead to trouble. Cybercrime is a sneaky kind of crime that happens especially often with technology. Hacking into important data has become more common and makes many people uneasy when using tech tools. That's why a system was created! Cryptography helps fight back against cybercrime by turning information into tricky codes

Citation: Rahul Das, Ashok Kumar Mahato, Suresh Kumar Sahani. Study on Computer Applications of Number Theory in Cryptography. Central Asian Journal of Mathematical Theory and Computer Sciences 2024, 5(3), 320-328.

Received: 27th July 2024
Revised: 27th August 2024
Accepted: 3th Sept 2024
Published: 10th Sept 2024

that are really tough for anyone who shouldn't see them to read. So, we can stay safer while enjoying all the neat things technology offers!

2. Materials and Methods

Bibliographic research is a popular way to find lots of detailed information. It's about collecting info from different places like books, journals, other helpful materials. Researchers can look at books and articles that relate to the question they are exploring. There are three main techniques they use to gather data: edit, organize, & analyze. Now, editing is checking the info to make sure it's clear and complete. Next up is organizing, which means putting everything into a neat framework.

Then comes analysis! This step digs deeper into the organized info using certain rules and theories [3]. The answers found help solve the problem being studied. When researchers analyze data, they often follow three steps: data reduction, data presentation, and data interpretation. Data reduction is about picking out the key ideas that match the main theme or pattern. For presenting data, it can be shown in charts, short descriptions, relationships between categories, flow diagrams, & other similar ways. If researchers find strong and consistent evidence during this process, their conclusions become very credible (which is good). So there you have it! Bibliographic research helps us learn so much more.

3. Results and Discussion

Number theory is a fun part of pure math. It looks closely at whole numbers which are also called integers. Whole numbers are those guys without any decimal points – like 8, 21, -5, or even 0!

Now real numbers do have decimals, like 8.0, 6.7, or 1.9. There's something interesting about integer division too [4]. If you have two integers, let's call them a and b (and a isn't zero), we can say that a divides b if we can find another integer, c, so that b equals a times c. We write this as (a | b) when b = ac with c being an integer & a not being zero.

The Euclidean theorems also belong in number theory. They're based on the idea that m and n are integers. When dividing m by n, you get q (the quotient) and r (the remainder). So it goes like this:

$$(m = nq + r)$$

and remember that r has to be between 0 and n ($0 \leq r \leq n$).

Let's chat about the Greatest Common Divisor (GCD). It's an important concept too! If you have two non-zero integers, a and b, the GCD is the biggest integer d that divides both of them [5]. You can write this as GCD(a, b) = d. When two integers have a GCD of 1, they're called relatively prime. Take GCD(36, 17) for instance; it equals 1! So, 36 and 17 are relatives in a math sense! This also connects to something called a linear combination. That means we can say:

$$(ma + nb = 1)$$

where m and n are some integers. And if our integers a and b are both positive.

In short: GCD(a, b) can totally be shown as (ma + nb). Isn't number theory neat?

Modulo arithmetic is super important in number theory. You know that value? That's the modulo. The results you when you do arithmetic with modulo m will always belong to the set {0, 1, 2, ..., m - 1}.

When you have two integers, a and m (with m being more than zero), you can do the operation a mod m (we say "a modulo m"). This gives you the remainder of dividing a by m. Basically, it means that if you write it like this: a mod m = r, then you can also express that as a = m * q + r, where $0 \leq r < m$. Remember, we call that value m the modulus or modulo.

Now, congruency is another term in number theory [6]. Let's say we have integers *a* and *b*. If they satisfy the rule where m > 0, then we can say that a ≡ b (mod m) if m divides (a - b). If they aren't congruent under modulus m, then we write it as,

$$a \not\equiv b \pmod{m}$$

Let's look at what happens when m is a positive integer.

1. If (a ≡ b) (mod m) and c is just any integer
    a. Then, (a + c) ≡ (b + c) (mod m)
    b. Also, ac ≡ bc (mod m)
    c. And if p is some non-negative integer, ap ≡ bp (mod m)
2. If a ≡ b (mod m) and c ≡ d (mod m), then:
    a. It follows that (a + c) ≡ (b + d) (mod m)
    b. And ac ≡ bd (mod m)

In regular number math, if you've got a non-zero number and want its inverse, it's simply a fraction that multiplies with the original number to give you back one. For instance, if we take a non-zero number like a, its inverse would be written as 1/a, because (a × a-1 = 1). We label this inverse as a-1.

But when it comes to modulo arithmetic, if you have an integer a and it's checked against mod m—if they're relatively prime and m > 1—then there's an inverse for a mod m. It's some integer x such that

$$ax \equiv 1 \pmod{m}$$

In another way of writing it:

$$a^{-1} \pmod{m} = x$$

The blend of congruence & linear combinations leads to what's called linear congruence. It's often shaped like this:

$$ax \equiv b \pmod{m}$$

with m being > 0; both a & b can be any integers while x is your integer variable.

You might run into k too! That's just an integer that helps round x numbers.

Now let's talk about the Euclidean algorithm—it's famous for finding the greatest common divisor of two integers [7]. Euclid came up with it long back, and he was this clever Greek mathematician. The trick involves repeating the division algorithm over & over.

To find GCD using the Euclidean Algorithm:

1. First off: when n equals zero, then guess what? GCD(m, n) = m, But if n isn't zero yet? You move on to step 2.
2. You'll divide m by n & remember that r is your remainder.
3. Now swap things around: take n for your new m and r for your new n; head back to step 1.

Just to illustrate this—let's say r0 = m and r1 = n then you'd see how it works:

$$r0 = r1 * q1 + r2 \text{ , with restrictions on r2 like: } 0 \le r2 < r1$$
$$r1 = r2 \,.\, q2 + r3 \text{ , } 0 \le r3 < r2$$
$$\vdots$$
$$rn\text{-}1 = rn\text{-}1 \,.\, qn\text{-}1 + rn \text{ , } \quad 0 \le rn < rn\text{-}1$$
$$n = rn \,.\, qn + 0$$

So you get:

$$GCD(m, n) = GCD(r0, r1) = GCD(r1, r2) = \ldots GCD(rn\text{-}1, rn) = GCD(rn, 0) = rn$$

Example : m = 525, n = 231 dan dipenuhi syarat m * n

$$512 = 2 * 231 + 63$$
$$231 = 3 * 63 + 42$$
$$63 = 1 * 42 + 21$$

$$42 = 2 * 21 + 0$$

So you get:

GCD(512, 231) = GCD(231, 63) = GCD(63, 42) = GCD(42, 21) = GCD(21, 0) = 21

The Euclidean algorithm is quite famous. It's named after a Greekian named Euclid, wrote about it in his book, Now, let's talk about cryptography. It plays a super important role in keeping information safe [8]. The word cryptography comes from two Greek words: "cryptos meaning secret, & "graphein" which means writing. Basically, cryptography a field that looks at math techniques to protect information. This includes things like confidentiality (keeping info private), integrity (making sure data isn't messed with), & authentication (proving who sent the message). Schneier describes it as both the science and art of keeping messages secure. So, it's about ensuring that messages stay private, the data remains intact, confirming who sent the info, & stopping denial from the sender.

In cryptography, there are lots of technical terms you'll hear. Here's a quick list of common ones:

1. Information can be understood through both sight and sound.
2. The sender is the person or group sending out the message.
3. The recipient is the one who gets the message.
4. Ciphertext refers to messages that get coded to keep them safe from people who shouldn't read them.
5. Encryption is how we change regular text into ciphertext.
6. Decryption is when we turn ciphertext back into its original text.
7. Cipher refers to methods used for encoding & decoding messages.
8. Keys are special bits used in encryption and decryption.
9. During sending messages, there can be sneaky eavesdroppers trying to intercept them.
10. Cryptoanalysis is all about cracking ciphertext back into regular text without knowing the key—those doing this are called cryptanalysts. An Arab scientist from the 9th century named Al-Kindi first suggested this idea.
11. Cryptology is simply the study of both cryptography & cryptoanalysis.

For thousands of years, cryptography has been part of our lives. Its roots go way back — way before today's tech became fancy and complicated! Ancient Egyptians were using it around 4000 years ago with unique hieroglyphs to send messages on pyramid walls [9].

Ancient civilizations like Egypt, Greece, Rome, & India their own ways of using cryptography. The Greeks were pretty clever; came up with a tool called the scytale about 400 years before Christ! There's a book called Arab Origins of Cryptology, made by the King Faisal Center in Saudi Arabia. It talks all about the Arab world's history with cryptography [10]. This ancient skill helped people send secret messages. For example, in the Kama Sutra, there's advice for gals on how to get the hang of ciphers. Two types mentioned are Kautiliyam & Mulavediy.

In Europe, folks also started using cryptography during the Renaissance—particularly around the 15th and 16th centuries. A bunch of codes became super popular back then:

1. The Vigenere Cipher kicked off in 1586, thanks to a French diplomat named Blaise de Vigenere.
2. The Playfair Cipher was pushed by a British diplomat called Lord Playfair but was originally invented by Charles Wheatstone way back in 1854.

But not everything about cryptography is great. It had a dark side too. In the 17th century, Queen Mary of Scotland got caught because Thomas Phelippes—a code breaker—figured out her secret message about taking down Queen Elizabeth I while she was stuck in prison. Even during World War II, cryptography played a big role. The German Nazis made a super-secret machine called Enigma [11]. Luckily, the Allies cracked that code! Solving Enigma is often seen as one reason World War II didn't last forever. Cryptography relies on algorithms that use number theory. There are lots of different cryptographic

algorithms out there like RSA, Elgamal, Diffie-Hellman key exchange, and knapsack algorithms. They can be split into three groups:

a. Symmetric Key Cryptography
b. Asymmetric Key Cryptography
c. Hash Function

Symmetric Key Cryptography (or Secret Key Cryptography): Is an old-school method where you use one key to encrypt and decrypt stuff! This method's been around for over 4,000 years. To read a message sent this way, you need that special key. So if you know the key, you can encrypt or decrypt messages easily [12].

Algorithms that use symmetric keys include:

a. RC2, RC4, RC5, RC6
b. One Time Pad (OTP)
c. Data Encryption Standard (DES)
d. Advanced Encryption Standard (AES)
e. Data Encryption Algorithm (IDEA)
f. A5, & others

Asymmetric key cryptography: This fancy term means using different keys for encryption & decryption. It's also known as public key cryptography. In this system, the key is split into two parts:

1. Public key: This key is okay for everyone to know! It gets published so anyone can use it.
2. Secret key (or private key): This one's kept under wraps. Only one person knows it!

These two keys are connected in a special way. A public key can encrypt messages, but here's the catch—you need the secret key to decrypt them. So, only the person with the secret key can read the message. Asymmetric algorithms are generally seen as more secure when sending messages compared to symmetric ones.

Algorithms that use of public keys include:

a. Digital Signature (DSA)
b. RSA
c. Diffie-Hellman (DH)
d. Elliptic Curve Cryptography (ECC)
e. Cryptography

Hash function: A hash function is a mathematical tool! It takes in input of any length and turns it into a fixed-length binary sequence. This is often called a one-way function or a message digest. Sometimes it's also known as a compression function or a Message Authentication Code (MAC). They act like a fingerprint for messages. Fingerprinting helps us check if a message is real and it ensures that nobody has messed with it.

Some of the common hash function include:

a. MD5
b. SHA-1, SHA-2, SHA-3
c. MAC

RSA Algorithm: RSA is a very popular public-key algorithm It does a lot important things, This first found by some smart folks: Rivest, Adi Shamir, and Adleman. Were working at MIT in 1976 they figured it out, The name RSA comes the first letters of their last names. So, what makes RSA safe? Well, it's all about figuring out large numbers. RSA's security comes from the hard job of breaking down a big number (we'll call it n) into two smaller prime numbers (let's say p and q). Here's n is made by multiplying p and q together! you can find another number using Ø(n) = (p-1) × (q-1). Now, if someone knows the encryption key e—and that's a big "if"—then they can find another key called the decryption key d. This works through a special math rule:

$$ed \equiv 1 \ (mod(n))$$

The person who created RSA suggests that both p and q should be more than 100 digits long. Because then their product, which we call $n = p \times q$, will have over 200 digits. something important: The fastest ways to factor numbers are pretty complicated. If you look at an integer like n that has b bits—the truth is—no quick way to break down those big numbers in polynomial time has been found yet. So right now, people still think RSA is secure. Long story short, the bigger the number, the harder it is to factor.

Even though the RSA algorithm has some flaws, it's really popular. It not as quick as other encryption methods like DES or AES. RSA used for sending secret messages directly Instead, it helps to send a symmetric key (which is like a session key) using the recipient's public key. The actual messages are encrypted using a symmetric method, like DES or AES. So, both the message and the symmetric key go together when using RSA. The person receiving them uses their private key to unlock the symmetric key first. After that, they can easily decrypt the message with that key.

Example-1:

Example demonstrating the operation of the RSA cryptosystem with small values.

Assume that n = 143 and Ø(n) = 120 result from p = 11 and q = 13. Given that GCD(7, 120) = 1, we can select e = 7.

Since 7*103 = 1 (mod 120), we can compute d = 103 using the extended Euclidean technique.

We have (143, 7) as our public key and (143, 103) as our private key.

Let's say we wish to encrypt the "HELLO" message.

Using ASCII encoding, we can translate this to the integer 726564766. We calculate the ciphertext as c = 726564766^7 (mod 143) = 32 using the public key.

Using the private key, we can decrypt the ciphertext by computing the original message, m = 32^103 (mod 143) = 726564766.

Example-2: If p = 7, q = 11, and e = 13 in the RSA algorithm, what will the value of d be?

Solution: Let p = 7, q = 11, and e = 13.

According to the RSA algorithm, Ø(n) = (7–1) × (11–1) = 6 × 10 = 60

Consequently, (e × d) mod Ø(n) = 1 (13 × d) mod 60 = 1 d = 37. Thus, key is 37.

Example-3: A participant in an RSA cryptosystem generates his public and private keys using the prime numbers p = 3 and q = 11. How will the text computer be encrypted using the public key if the private key is 7?

Solution: Given prime integers, n = 3 x 11 = 33, p = 3, and q = 11.

Ø(n) is equal to (3-1) × (11-1) = 2 x 10 = 20

In response to question d = 7, the greatest common divisor (20, d) = 1.

Mod Ø(n) = (e × d) = 1.

(e x 7) mod 20 = 1

Consequently, e x 7= 20 x 1+1

e = 217 = 3 is feasible.

Encrypt key: e = public key = 3

Thus, n = 33, e = 3, d = 7, and Ø(n) = 20.

Computer is the plan text. m mod n is the ciphertext.

For C = 33 mod 33 = 27, the ciphertext

For O = 153 mod 33 = 9, the ciphertext

Given prime integers, n = 3 x 11 = 33, p = 3, and q = 11.

Ø(n) is equal to (3-1) × (11-1) = 2 x 10 = 20

In response to question d = 7, the greatest common divisor (20, d) = 1.

Mod Ø(n) = (e × d) = 1.

(e x 7) mod 20 = 1

Consequently, e x 7= 20 x 1+1

e = 217 = 3 is feasible.

Similarly,

For M = 133 mod 33 = 19, the ciphertext
For P = 163 mod 33 = 4, the ciphertext
For U = 213 mod 33 = 21, the ciphertext
For T = 203 mod 33 = 14, the ciphertext
For E = 53 mod 33 = 26, the ciphertext
For R = 183 mod 33 = 24, the ciphertext

The Elgamal algorithm: It was made by Taher Elgamal back in 1985 and first appeared in a paper called "A public key cryptosystem and a signature scheme based on discrete logarithms." Its security comes from how tough it is to solve discrete logarithms. You run into this problem when p is prime, and g & y are any integers, with x needing to be found, looking something like this:

$$g^x \equiv y \pmod{p}$$

Here's how encryption works in the Elgamal algorithm:

1. Break the plaintext into blocks.
2. Pick a random number k that fits: $1 \leq k \leq p - 2$.
3. Each block gets encrypted using this formula:

$$a = g^k \pmod{p}$$
$$b = y^k m \pmod{p}$$

The pair of values a & b represents the encrypted message block. This means that the ciphertext is actually double the size of the plaintext. The decryption process includes the following:

1. Use your private key x to calculate $(a^x)^{-1} \equiv a^{p-x-1}$.
2. Then calculate plaintext using this equation:

$m = b / (a^x) \pmod{p}$, which is also written as $m = b(a^x)^{-1} \pmod{p}$

Example: Sonam chooses $p_A = 107$, $\alpha_A = 2$, $d_A = 67$, and she computes $\beta_A = 267 \equiv 94 \pmod{107}$. Her public key is $(p_A, \alpha_A, \beta_A) = (2,67,94)$, and her private key is $d_A = 67$.

Rahul wants to send the message "B" (66 in ASCII) to Sonam. He chooses a random integer k = 45 and encrypts M = 66 as $(r, t) = (\alpha_A{}^k, \beta_A{}^k M) \equiv (245, 944566) \equiv (28, 9) \pmod{107}$. He sends the encrypted message (28, 9) to Sonam.

Sonam receives the message (r, t) = (28, 9), and using her private key $d_A = 67$ she decrypts to $tr^{-dA} = 9 \cdot 28^{-67} \equiv 9 \cdot 28106^{-67} \equiv 9 \cdot 43 \equiv 66 \pmod{10}$.

Diffie Hellman key Exchange Algorithm: Whitfield Diffie & Martin Hellman created the Diffie-Hellman key exchange method. This algorithm allows two people to share a secret key safely for encrypting messages with symmetric algorithms like DES and AES. Just like Elgamal, it relies on how tricky it is to compute discrete logarithms for security.

Knapsack cryptographic algorithm (or Merkle-Hellman algorithm): Ralph Merkle & Martin Hellman discovered it in 1978, making it one of the early public key methods. It's based on something called the knapsack problem, which is really well-known in computer science because it can't be solved quickly by any fast method. This algorithm turns messages into a series of solutions to that problem. Here, each weight (wi) in the knapsack acts as a secret key while plaintext bits are represented as bi.

The AES Algorithm: It's an encryption method that works with just one key for both encryption and decryption—a symmetric approach! AES has three options for keys: AES-128, AES-192, and AES-256, each round uses a different internal key called the round key for each round process. For AES-128 specifically, there are ten rounds:

1. First comes AddRoundKey.
2. Nine times you repeat SubstituteBytes, ShiftRows, MixColumns & AddRoundKey.
3. Finally, in the last round you simply perform SubBytes, ShiftRows &, AddRoundKey.

During decryption for AES-128, You go through that rotation process 10 times. Specifically, it is carried out as follows :

1. Round Key Add
2. Every 1 of the 9 rounds involves the following procedures being completed: ReverseShift Rows, AddRoundKey, InverseMixColumns, and InverseSubstituteBytes.
3. The last round consists of the InverseSubBytes, InverseShiftRows, and AddRoundKey methods.

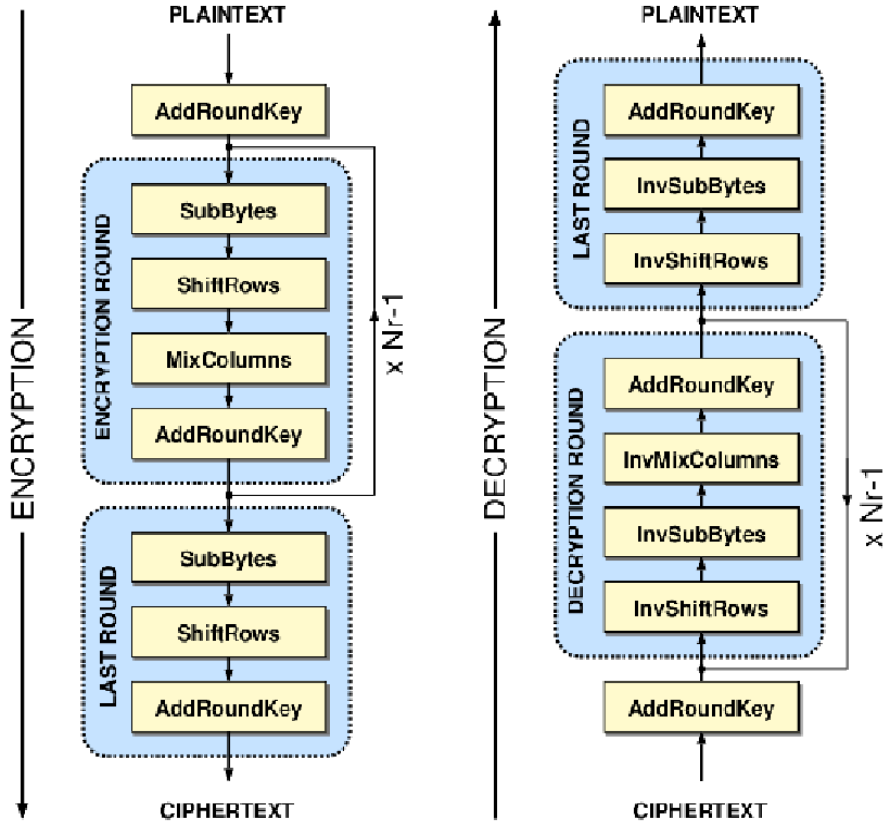

Figure 1. Concept of AES-128 Encryption and Decryption

4. Conclusion

Numerous mathematical applications, including cryptography methods, stem from the field of number theory. The study of converting useful information into meaningless information and vice versa is known as cryptography. It uses particular encryption techniques or algorithms to keep messages private. Because it increases security and prevents illegal data or information disclosure, cryptography is essential to technology. Numerous cryptographic techniques, including the Diffie-Hellman key exchange, Elgamal, RSA, and knapsack cryptographic algorithms, are based on number theory. Since cryptography seeks to improve security and prevent unauthorized data or information leakage, it is a useful tool in technology. For cryptography, there are many different algorithms accessible. Depending on its intended usage, the best method should be chosen.

Acknowledgments: It gives me great pleasure to thank Er. Dhananjaya Mandal and my mentor, Suresh Kumar Sahani, for their kind assistance and encouragement. Without them, I could not have chosen such an intriguing topic or completed my paper.

REFERENCES

[1] [Online]. "ElGamal Encryption Algorithm," GeeksforGeeks. Available: https://www.geeksforgeeks.org/elgamal-encryption-algorithm/. Accessed: Dec. 12, 2022.

[2] Prayitno and N. Nurdin, "Analisa dan Implementasi Kriptografi pada Pesan Rahasia," J. Elektron. Sist. Inf. dan Komput., vol. 3, no. 1, pp. 1–11, 2017. [Online]. Available: nnurdin69@gmail.com.

[3] "AES - Advanced Encryption Standard," Study Informatics, Jul. 2012. [Online]. Available: http://studyinformatics.blogspot.com/2012/07/aes-advanced-encryption-standard.html. Accessed: Dec. 12, 2022.

[4] "Algoritma RSA," KomputerKata, 2012. [Online]. Available: https://komputerkata.com/algoritma-rsa/. Accessed: Dec. 12, 2022.

[5] D. Ariyus, Kriptografi: Keamanan Data dan Komunikasi. Yogyakarta: Graha Ilmu, 2006.

[6] I. M. A. Arrijal, R. Efendi, and B. Susilo, "Penerapan Algoritma Kriptografi Kunci Simetris dengan Modifikasi Vigenere Cipher dalam Aplikasi Kriptografi Teks," J. Pseudocode, vol. 3, no. 1, pp. 69–82, 2016. [Online]. Available: https://doi.org/10.33369/pseudocode.3.1.69-82.

[7] B. S. Hasugian, "Peranan Kriptografi Sebagai Keamanan Sistem Informasi pada Usaha Kecil dan Menengah," J. Chem. Inf. Model., vol. 53, no. 9, pp. 1689–1699, 2019. [Online]. Available: https://doi.org/10.46576/wdw.v0i53.269.

[8] E. Endaryono, N. Dwitiyanti, and H. S. Setiawan, "Aplikasi Operasi Matriks pada Perancangan Simulasi Metode Hill Cipher Menggunakan Microsoft Excel," STRING, vol. 6, no. 1, pp. 1–9, Aug. 2021. [Online]. Available: http://dx.doi.org/10.30998/string.v6i1.8603.

[9] A. M. T., B. Konseling, F. I. Pendidikan, and U. N. Surabaya, "Studi Kepustakaan Mengenai Landasan Teori dan Praktik Konseling Expressive Writing," J. BK UNESA, vol. 8, pp. 1–8, 2018. [Online]. Available: https://ejournal.unesa.ac.id/index.php/jurnal-bk-unesa/article/view/22037.

[10] "What is Cryptography: Types, Tools and Its Algorithms," Elprocus. [Online]. Available: https://www.elprocus.com/cryptography-and-its-concepts/. Accessed: Dec. 11, 2022.

[11] "What Is Encryption?," Cisco. [Online]. Available: https://www.cisco.com/c/en/us/products/security/encryption-explained.html. Accessed: Dec. 11, 2022.

[12] W. Widaya, Modul Penyusunan Soal Keterampilan Berpikir Tingkat Tinggi (Higher Order Thinking Skills) Matematika. Jakarta: Direktorat Pembinaan Sekolah Menengah Atas, 2018.