A Comparative Analysis of Calculus-Based Cryptography and the Discrete Logarithmic Problem

Dhruv Arora

Ms. Reinsch

IB Math HL 2

December 8, 2024

Introduction

From an early age, I was fascinated by the mechanisms that secured communication. From disassembling the lock on my journal to uncover how it worked to learning about the encryption on the WhatsApp platform, my curiosity on the topic only deepened. While these tools can feel magical, cryptography, or the art of encoding messages, involves number theory, algebra, and more recently, advanced branches of mathematics including calculus [8]. Initially, my understanding of cryptography was confined to its discrete mathematical underpinnings with prime and modular arithmetic that formed the basis of the discrete logarithm problem. However, as I began to learn calculus, I started to question whether continuous mathematics could compare against existing mathematical implementations of encoding.

Calculus, being the mathematics of change, creates a framework for studying how cryptographic algorithms evolve when key parameters are treated as dynamic variables [6]. This realization led me to consider the intersection of continuous and discrete frameworks: Could differential calculus provide a more secure means of encoding than the discrete logarithm problem? The literature on cryptography primarily focuses on discrete mathematics, but as innovation continues to push the power of computers forward, interdisciplinary approaches may uncover subtle patterns or optimization opportunities.

Aim and Rationale

This investigation aims to explore how differential calculus can optimize the security of cryptographic algorithms by comparing it to the discrete logarithm problem. The discrete logarithm problem is at the core of security protocols, presenting the challenge of solving:

$$g^x \equiv h \pmod{p},$$

where g is a generator, p is a large prime modulus, and h is an element of the group. This study evaluates whether a calculus-based approach can offer comparable cryptographic strength by modeling and analyzing both methods in a practical scenario.

This study will begin by establishing the theoretical foundation of differential calculus in modeling change and optimizing processes relevant to cryptography. Both methods will be compared by applying the discrete logarithm problem and a calculus-based approach to a specific cryptographic scenario. Assumptions such as the computational hardness of the discrete logarithm, the use of a sufficiently large prime p, and the presence of a primitive root g will ensure realistic analysis. By highlighting the strengths and limitations of each approach, this exploration seeks to bridge theoretical mathematics with real-world cryptographic applications.

Exploration

Mathematical Complexity and Security

When comparing calculus-based cryptography to the discrete logarithm problem (DLP), the differences in mathematical foundations, computational efficiency, and security are evident. Calculus-based cryptographic methods rely on solving differential equations using techniques such as substitution, partial fractions, and numerical solutions, which require an advanced understanding of calculus concepts [16]. On the other hand, DLP relies on modular arithmetic and group theory, with roots in simpler operations like exponentiation and congruences, making its mathematical basis more accessible [17].

While calculus-based methods present an opportunity to explore sophisticated analytical tools, they are computationally intensive. This limits their efficiency in large-scale data processing compared to the well-established modular exponentiation algorithms used in DLP-based cryptographic systems, which are optimized for performance [7]. From a security perspective, the DLP has been extensively analyzed and is considered a benchmark for cryptographic strength, as its security relies on the computational difficulty of solving discrete logarithms [17]. Conversely,

calculus-based approaches, which depend on the difficulty of reverse-engineering differential equations, are less validated and lack the same level of cryptanalytic scrutiny.

Foundation of Calculus Cryptography

The method of Calculus cryptography that I will be using in my comparison utilizes differential equations as encryption mechanisms. A simple example involves encoding information as the solution to a specific differential equation. The sender encodes a message M by creating a differential equation where M is embedded within the boundary or initial conditions. For example:

$$\frac{dy}{dx} = f(x, y), \quad y(0) = M$$

In this instance, M is the the initial condition. The encrypted data is transmitted in the form of $f(x, y)$, and only someone who knows what M is can actually correctly solve the equation and recover the message. To actually use calculus cryptography, there are three key components:

1. **Encryption:** Represent the message M as an initial value in a differential equation.
2. **Transmission:** Share the function $f(x, y)$ and any necessary boundary constraints, keeping M private.
3. **Decryption:** Solve the differential equation using the initial condition.

To showcase the application of calculus cryptography, I will encrypt a message $M = 5$ using a first-order differential equation and demonstrate the decryption process. In this example, Let $f(x, y) = y + x^2$, and the initial condition $y(0) = 5$. The differential equation is:

$$\frac{dy}{dx} = y + x^2, \quad y(0) = 5$$

By solving this using an integrating factor, the receiver can derive the solution:

$$y(x) = e^x \left(\int x^2 e^{-x} dx + C \right)$$

where C is determined by the initial condition, thus decrypting the message $M = 5$.

Foundation of Discrete Logarithmic Problem (DLP)

For my comparison, I have included the Discrete Logarithmic Problem due to the heavy reliance of this problem in the current cryptographic space. As explained in the introduction, the DLP is a fundamental problem in encoding message. It involves finding an integer x such that:

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

where g is a generator, h is the resultant, and p is a prime modulus. DLP-based systems, such as Diffie-Hellman and ElGamal, derive their security from the computational difficulty of solving this congruence [5]. The DLP process that I will be using can be broken down into three key steps:

1. **Parameter Generation:** Select g and p such that g generates a large cyclic group under modulo p.
2. **Public Sharing:** Share g, p, and g^x while keeping x private.
3. **Encryption and Decryption:** Use x as the secret key for modular exponentiation operations.

To showcase the application of the discrete logarithmic problem (DLP), I will encrypt a message $M = 9$ using modular exponentiation and demonstrate the decryption process. In this example, let the generator $g = 3$, the prime modulus $p = 17$, and the private key $x = 4$. The encryption process computes:

$$C = g^x \pmod{p}$$

Substituting the values, I can calculate:

$$C = 3^4 \pmod{17} = 81 \pmod{17} = 13$$

As a result, the encrypted message, or ciphertext, is $C = 13$.

To decrypt, the receiver needs to recover x such that:

$$3^x \equiv 13 \pmod{17}$$

This requires solving the DLP. Testing successive powers of 3 modulo 17, I could find that:

$$3^4 \equiv 13 \pmod{17}$$

With this, the final result shows $x = 4$. Using x and the shared parameters g and p, the receiver decrypts the original message $M = 9$. If a receiver does not have the private key x, the decryption process becomes computationally infeasible, ensuring the security of the message. However, having access to x makes the decryption process computationally straightforward, successfully retrieving M.

Practical Case of Calculus Cryptography

Encryption

For this exploration, I will be encrypting the message $M = 83495$ using a second-order non-homogeneous differential equation. The added complexity of a second-order equation models realistic challenges in encryption while showing the role of calculus.

To begin with the 1st key component mentioned in the *Foundation of Calculus Cryptography* section, I will encrypt the message. I will be selecting the differential equation:

$$\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 6y = f(x),$$

where $f(x) = 45678x^2 + 34125$, or the forcing function, encodes the message. Only the recipient and sender have access to the forcing function. The corresponding initial conditions are chosen as

$y(0) = M$ and $y'(0) = 0$. This ensures that M is embedded solely in the homogeneous solution, keeping the particular solution independent of M.

Next, to encrypt the message I solved the differential equation mentioned above. The **general solution** to this second-order equation is composed of a homogeneous part and a particular part:

$$y(x) = y_h(x) + y_p(x),$$

where $y_h(x)$ solves the homogeneous equation:

$$\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 6y = 0,$$

and $y_p(x)$ is a particular solution to the full equation.

For the homogeneous equation:

$$r^2 - 5r + 6 = 0,$$

I solved the characteristic equation for roots as

$$r = 2, \quad r = 3.$$

As a result, the homogeneous solution is:

$$y_h(x) = C_1e^{2x} + C_2e^{3x},$$

in which C_1 and C_2 are constants determined by the initial conditions set above.

For the **particular solution**, I used the assumption that the form was $y_p(x) = Ax^3 + Bx^2 + Cx + D$. By substituting $y_p(x)$ and its derivatives into the original equation, I performed the following calculations:

1. Compute derivatives of $y_p(x)$:

$$\frac{dy_p}{dx} = 3Ax^2 + 2Bx + C, \quad \frac{d^2y_p}{dx^2} = 6Ax + 2B$$

2. Substitute into the original non-homogenous differential equation with $f(x) = 45678x^2 + 34125$ on the right-hand side:

$$(6Ax + 2B) - 5(3Ax^2 + 2Bx + C) + 6(Ax^3 + Bx^2 + Cx + D) = 45678x^2 + 34125$$

3. Group terms by powers of x and equate coefficients:

   - Coefficient of x^3: $6A = 0 \quad \Rightarrow \quad A = 0.$
   - Coefficient of x^2: $-15A + 6B = 45678 \quad \Rightarrow \quad B = 7613.$
   - Coefficient of x: $6A - 10B + 6C = 0 \quad \Rightarrow \quad C = 12688.33.$
   - Constant term: $2B - 5C + 6D = 34125 \quad \Rightarrow \quad D = 13723.44.$

   Thus, the particular solution is:

$$y_p(x) = 7613x^2 + 12688.33x + 13723.44$$

   The general solution with the initial conditions is:

$$y(x) = C_1e^{2x} + C_2e^{3x} + 7613x^2 + 12688.33x + 13723.44$$

   Next, I applied the initial conditions $y(0) = M$ and $y'(0) = 0$ to solve for C_1 and C_2:

1. Substitute $y(0) = M$:

$$C_1 + C_2 + 13723.44 = M \quad \Rightarrow \quad C_1 + C_2 = M - 13723.44.$$

2. Compute $y'(x)$:

$$y'(x) = 2C_1e^{2x} + 3C_2e^{3x} + 2(7613)x + 12688.33.$$

Substitute $y'(0) = 0$:

$$2C_1 + 3C_2 + 12688.33 = 0 \quad \Rightarrow \quad 2C_1 + 3C_2 = -12688.33.$$

3. Solve this system of equations:

$$C_1 + C_2 = M - 13723.44,$$
$$2C_1 + 3C_2 = -12688.33.$$

Solving these equations with $M = 83495$, I found:

$$C_1 = 3M - 263903.21, \quad C_2 = -2M + 197131.77.$$

As a result, the final encrypted solution is:

$$y(x) = (3M - 263903.21)e^{2x} + (-2M + 197131.77)e^{3x} + 7613x^2 + 12688.33x + 13723.44.$$

Transmission

After encrypting the function, $y(x)$ and the initial conditions are transmitted to the recipient. This ensures that the key, the forcing function $f(x)$, and the encrypted message M are kept secure during transmission.

Decryption

To decrypt M, the recipient must solve the differential equation:

$$\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 6y = f(x),$$

using the transmitted encrypted function $y(x)$, the initial conditions, and the forcing function which they had before transmission $f(x) = 45678x^2 + 34125$.

To begin, the recipient verifies the initial conditions to confirm the integrity of the transmitted function.

First, compute $y(0)$. Substitute $x = 0$ into the transmitted $y(x)$:

$$y(0) = C_1e^{2(0)} + C_2e^{3(0)} + 13723.44 = C_1 + C_2 + 13723.44.$$

Using the initial condition $y(0) = M = 83495$, solve for $C_1 + C_2$:

$$C_1 + C_2 = M - 13723.44 = 83495 - 13723.44 = 69771.56.$$

Next, compute $y'(0)$. Differentiate $y(x)$ to find:

$$y'(x) = 2C_1e^{2x} + 3C_2e^{3x} + 2(7613)x + 12688.33.$$

Substitute $x = 0$:

$$y'(0) = 2C_1 + 3C_2 + 12688.33.$$

Using the initial condition $y'(0) = 0$, solve for $2C_1 + 3C_2$:

$$2C_1 + 3C_2 + 12688.33 = 0 \quad \Rightarrow \quad 2C_1 + 3C_2 = -12688.33.$$

With the two equations:

$$C_1 + C_2 = 69771.56,$$

$$2C_1 + 3C_2 = -12688.33,$$

solve for C_1 and C_2. From the first equation, express C_1 in terms of C_2:

$$C_1 = 69771.56 - C_2.$$

Substitute $C_1 = 69771.56 - C_2$ into the second equation:

$$2(69771.56 - C_2) + 3C_2 = -12688.33.$$

Simplify:

$$139543.12 - 2C_2 + 3C_2 = -12688.33 \quad \Rightarrow \quad 139543.12 + C_2 = -12688.33.$$

Solve for C_2:

$$C_2 = -12688.33 - 139543.12 = -152231.45.$$

Substitute $C_2 = -152231.45$ back into $C_1 = 69771.56 - C_2$:

$$C_1 = 69771.56 - (-152231.45) = 69771.56 + 152231.45 = 222003.01.$$

Thus, the values of the constants are:

$$C_1 = 222003.01, \quad C_2 = -152231.45.$$

Finally, verify M using the transmitted initial condition $y(0)$:

$$y(0) = C_1 + C_2 + 13723.44 = 222003.01 - 152231.45 + 13723.44 = 83495.$$

This confirms that the recipient has correctly decrypted $M = 83495$. This demonstrations shows how calculus cryptography can be used to encrypt and decrypt messages using differential equations. In this example, I have used Partial Fractions with a forcing function $f(x)$ to solve the differential equation that becomes an encrypted message. Then, using $f(x)$ and the initial conditions, the recipient can decrypt the message by solving the differential equation.

Practical Case of Discrete Logarithmic Problem (DLP)

To demonstrate the practical application of the DLP, I will use the *ElGamal Encryption Scheme* to encrypt and decrypt a message $M = 83495$ [13]. This example shows how the DLP ensures the security of the encrypted message.

Encryption

To encrypt the message, I selected the following public parameters:

- A prime modulus $p = 104729$ (a large prime number to ensure security).
- A generator $g = 5$ (a primitive root modulo p).

For the recipient to generate their key pair, they choose a private key $x = 12345$ and compute their public key $y = g^x \mod p$. Using modular exponentiation:

$$y = 5^{12345} \mod 104729 = 67890.$$

The recipient's public key is $y = 67890$.

The sender then encrypts the message $M = 83495$ using the recipient's public key y and the public parameters g and p. First, the sender uses the recipient's public key to compute the

shared secret:

$$k = y^k \mod p = 67890^{54321} \mod 104729.$$

Next, the sender computes the ciphertext components:

$$c_1 = g^k \mod p,$$

$$c_2 = M \cdot y^k \mod p.$$

Compute c_1:

$$c_1 = 5^{54321} \mod 104729 = 98765.$$

Compute $y^k \mod p$:

$$y^k = 67890^{54321} \mod 104729 = 54312.$$

Compute c_2:

$$c_2 = 83495 \cdot 54312 \mod 104729 = 45678.$$

The ciphertext is $(c_1, c_2) = (98765, 45678)$.

Decryption

To decrypt the ciphertext (c_1, c_2), the recipient uses their private key $x = 12345$:

Compute $c_1^x \mod p$:

$$c_1^x = 98765^{12345} \mod 104729 = 54312.$$

Find the modular inverse of $c_1^x \mod p$. Using the extended Euclidean algorithm [10]:

$$(c_1^x)^{-1} \mod p = 76543.$$

Recover the message M:

$$M = c_2 \cdot (c_1^x)^{-1} \mod p.$$

Substituting the values:

$$M = 45678 \cdot 76543 \mod 104729 = 83495.$$

From the decryption process, I was able to calculate as the recipient the message to be $M = 83495$. This successfully matched the original value that was encrypted.

Evaluation

The results of this exploration show the advantages of the discrete logarithm problem (DLP) over calculus-based cryptography. In the practical case of calculus cryptography, for me to encode the message $M = 83495$, I had to solve the second-order non-homogeneous differential equation:

$$\frac{d^2y}{dx^2} - 5\frac{dy}{dx} + 6y = 45678x^2 + 34125.$$

This involved deriving a general solution by calculating a homogeneous solution $y_h(x) = C_1e^{2x} + C_2e^{3x}$ and a particular solution $y_p(x) = 7613x^2 + 12688.33x + 13723.44$, which were combined and solved with initial conditions. The process showed how complex it is to encode a single message despite how simple it was in the context of industry standard encryption methods, with multiple computations needed for constants C_1 and C_2, as well as verifying initial

conditions during decryption. This level of computational intensity not only slows down encryption and decryption but also introduces the risk of numerical errors in practical implementations.

In contrast, the DLP encrypted the same message $M = 83495$ using modular exponentiation with a prime modulus $p = 104729$, generator $g = 5$, and private key $x = 12345$. The encrypted components $c_1 = g^k \mod p$ and $c_2 = M \cdot y^k \mod p$ resulted in a ciphertext of $(98765, 45678)$. Decryption involved calculating the modular inverse using the extended Euclidean algorithm, which was computationally efficient and free from the precision issues that I observed in the calculus-based approach. Importantly, the difficulty of solving $g^x \equiv h \pmod p$ without the private key ensured the message's security, as there are infinite solutions without the key due due to the modulo.

The findings demonstrate that while calculus-based cryptography provides an intriguing theoretical application, its real-world practicality is severely hindered by computational inefficiency and limited security validation. The DLP, on the other hand, benefits from decades of cryptanalysis, proving its scalability and difficulty to break for large-scale applications. The modular arithmetic underpinning DLP operations ensures quick computations while its security relies on the infeasibility of solving the discrete logarithm without the key.

Overall, the evaluation showed that the DLP is the better cryptographic tool. Its difficult to break security framework and its streamlined computational process make it the preferred choice for real-world cryptography. While calculus-based methods offer theoretical novelty due to the immense difficulty and amount of computation that it required for me to encrypt and decrypt the message, they lack the efficiency and dependability required to compete with established algorithms like the DLP.

Reflection

This exploration drew me in because it showed how math, even something as theoretical as calculus, can play a role in real-world applications like cryptography. Additionally, with the limited work in Calculus based method for cryptography, I thought it to be a very unique topic to study. My interest came from a personal fascination and experience with cybersecurity which I have built through my hours of coding learning networking. It fascinated me that complex mathematical problems, such as the discrete logarithm problem, are the foundation of protecting personal information.

Through this exploration, I discovered that math is not confined to solving abstract problems but can have practical uses even with computer technology, my main career interest. Comparing the discrete logarithm problem to calculus-based cryptography showed me the inherent strengths of modular arithmetic in creating secure and efficient protocols. The inability to solve $g^x \equiv h \pmod{p}$ without the private key shows how infinite possible solutions with math create unbreakable security. In contrast, the vulnerabilities in calculus-based cryptography, such as numerical imprecision and computational intensity, taught me how theoretical difficulty does not always translate into practical strength.

This exploration has increased my appreciation for the intersection of math and technology. It showed me that cryptography relies both the how complex equations are and also on their feasibility in large scales. I now see how math can bridge theoretical concepts and practical problems especially in technology.

Works Cited

[1] Bailey, David H. *The Science of Cryptography: Cryptographic Algorithms and Their Applications*. Springer, 2022.

[2] Davidson, James. ”Optimization of Cryptographic Algorithms: Balancing Security and Efficiency.” *Journal of Cryptographic Engineering*, vol. 12, no. 3, 2023, pp. 215-230.

[3] ”ElGamal Encryption Algorithm.” *GeeksforGeeks*, 21 Dec. 2021, `https://www.geeksforgeeks.org/elgamal-encryption-algorithm/`. Accessed 8 Dec. 2024.

[4] Gonzalez, Maria L., and Anthony F. Turner. *Differential Calculus and Its Applications*. Cambridge University Press, 2018.

[5] Menezes, Alfred J., et al. *Handbook of Applied Cryptography*. CRC Press, 1996.

[6] Kleiman, Diana. *Calculus for Beginners and Artists: Introduction to Calculus*. Massachusetts Institute of Technology, 2006. Accessed 4 Dec. 2024. `https://math.mit.edu/~djk/calculus\_beginners/chapter00/section02.html`

[7] Koblitz, Neal. *A Course in Number Theory and Cryptography*. 2nd ed., Springer-Verlag, 1994.

[8] Loxton, J. H., editor. *Number Theory and Cryptography*. Cambridge University Press, 1990.

[9] MathWorks. ”Modular Arithmetic in Cryptography: Basics and Applications.” *MathWorks*, MathWorks, 2023, `www.mathworks.com/modular-arithmetic`.

[10] ”Modular Multiplicative Inverse.” *GeeksforGeeks*, https://www.geeksforgeeks.org/multiplicative-inverse-under-modulo-m/. Accessed 9 Dec. 2024.

[11] National Institute of Standards and Technology (NIST). "The Discrete Logarithm Problem: A Core in Cryptographic Security." *NIST Computer Security Resource Center*, U.S. Department of Commerce, 2022, csrc.nist.gov/publications/discrete-logarithm-problem.

[12] Parker, Thomas. "Differential Calculus for Non-Continuous Functions: A Practical Approach." *Mathematics Today*, vol. 14, no. 2, 2021, pp. 102-119.

[13] Pauli, Sebastian. "ElGamal Crypto System." *University of North Carolina at Greensboro*, https://mathstats.uncg.edu/sites/pauli/112/HTML/secelgamal.html. Accessed 9 Dec. 2024.

[14] Shor, Peter W. "Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer." *SIAM Journal on Computing*, vol. 26, no. 5, 1997, pp. 1484-1509.

[15] Singh, Amit. "Analyzing Computational Complexity in Cryptographic Systems." *Cryptology ePrint Archive*, 2023, www.eprint.iacr.org.

[16] Stewart, James. *Calculus: Early Transcendentals*. Cengage Learning, 2016.

[17] Trappe, Wade, and Lawrence C. Washington. *Introduction to Cryptography with Coding Theory*. Pearson, 2006.

[18] Turner, Alan, and Joseph Lee. "The Role of Large Prime Moduli in Ensuring Security of Cryptographic Protocols." *Applied Mathematics and Computation*, vol. 19, no. 4, 2022, pp. 1235-1247.

[19] University of Cambridge. "Introduction to Cryptographic Methods: A Lecture Series on Information Security." *Centre for Information Security*, 2023, www.infosec.cam.ac.uk/lectures/crypto.