Optimizing a Window Method Approach for Shortest Addition Chain with Applications to the Elliptic Curve Cryptosystem

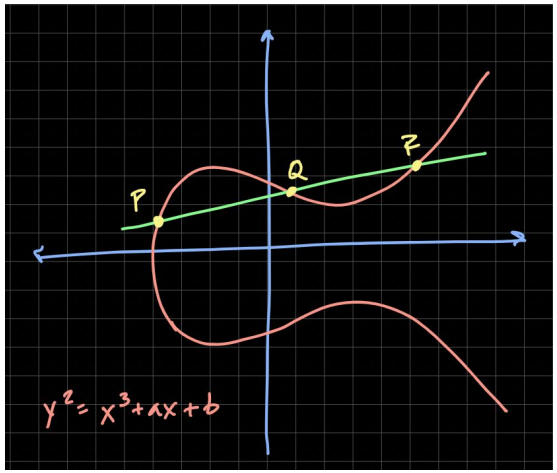


Figure 1 - "A Elliptic Curve in a Coordinate Plane"
Source: https://matt-rickard.com/elliptic-curve-cryptography

Project ID: 5110

Category: MA

Division: Senior

Math/Computer Science Type

Introduction / Purpose

What is Elliptic Curve Cryptography?

- Elliptic Curve RSA cryptography (ECC) is a common cryptosystem used for protecting data.
- Elliptic Curve RSA involves large computations of points on the elliptic curve.
- The time required for this computation is determined by the length of an “addition chain.”

Research Question

How can the Window Method be optimized to make ECC implementations more efficient?

Purpose

To find the **shortest addition chain** of a select value to further increase the speed of encryption.

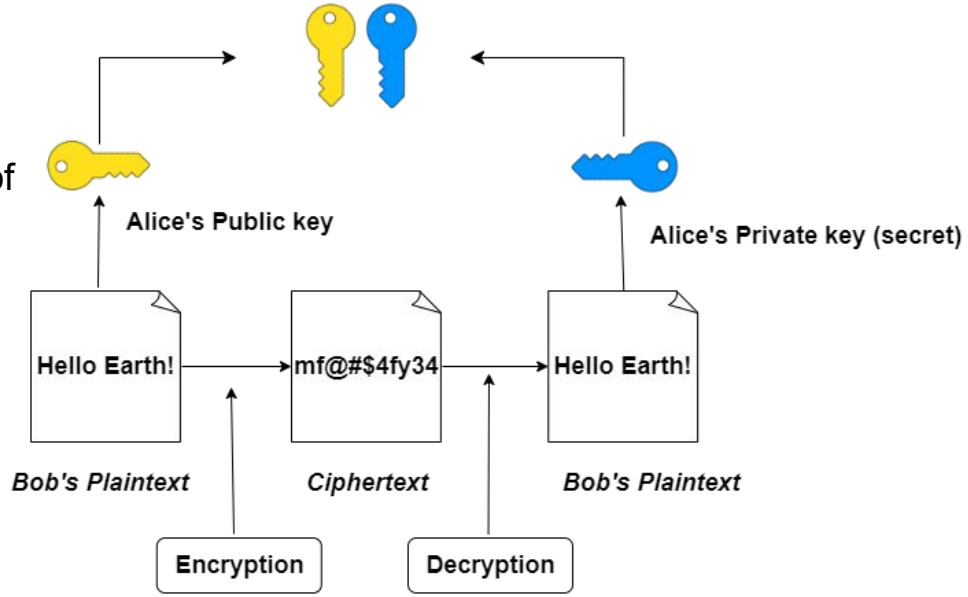


Figure 2 - A diagram of a common type of Cryptography: Public Key Encryption

Overview

Context: What is an Addition Chain?

Definition: A sequence of integers beginning with 1 such that each term in the sequence is the sum of any previous two terms in the sequence (can be the same term twice).

- Example: 1,2,3,4 (alternative chain for 4)
  - 1+1= 2, 2+1 = 3, 3+1 = 4
- Example: 1,2,4 **(shortest chain for 4)**
  - 1+1 = 2, 2+2 = 4
- Non-example 1,4 (not a addition chain)
  - 1+1≠ 4

The examples show chains of 4. The goal is to find the shortest addition chain to reach a target number.

Goal

1. Create **shortest addition chain** algorithm via the Window Method approach
2. Optimize window size use for such algorithm
3. Use the chain for providing a path on the Elliptic Curve for encryption

Finding the shortest addition chain allows the ECC system to encrypt/decrypt quickly. In general, finding the shortest addition chain is an NP-Hard Problem. This project aims to find the most efficient window size to produce the shortest addition chain.

Methodology: Using The Window Method

- The Window Method is a way of calculating the shortest addition chain
  - It looks at the binary sequence of a integer and divides into “windows.”
    - depending on the window size it doubles X times
    - then it adds the window to the value
- Window Size of 3 and 2 are used in this example.

Why the Window Method?

- It was chosen to see how it **be further optimized**

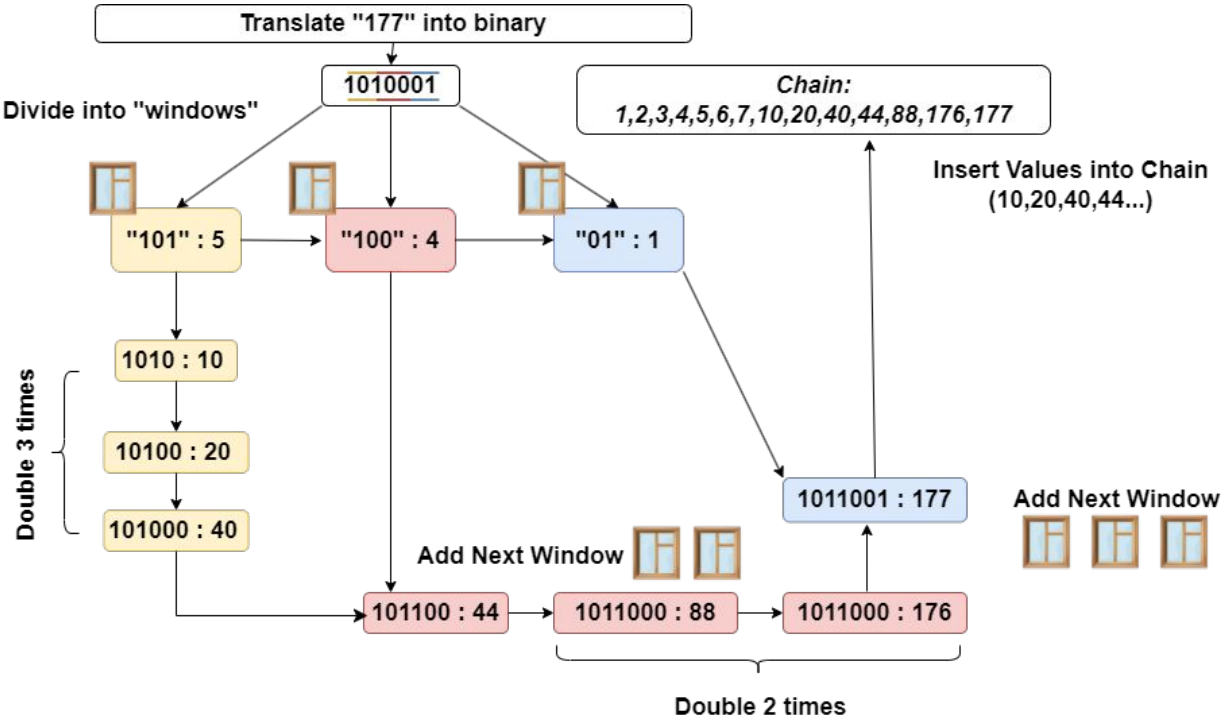


Figure 3 - An Example of Window Method Approach with 177

Methodology: Implementing the Algorithm

- Original coded developed in C++
- Algorithm was programmed based on preliminary pseudocode found in research stage (Kunihiro and Yamamoto)

Variables

"total" **-** a variable that shows the numerical value of all windows
"windowmultiplier" **-** a variable to store the multiplication of each window
"windowval" **-** a variable to store the numerical value of each window

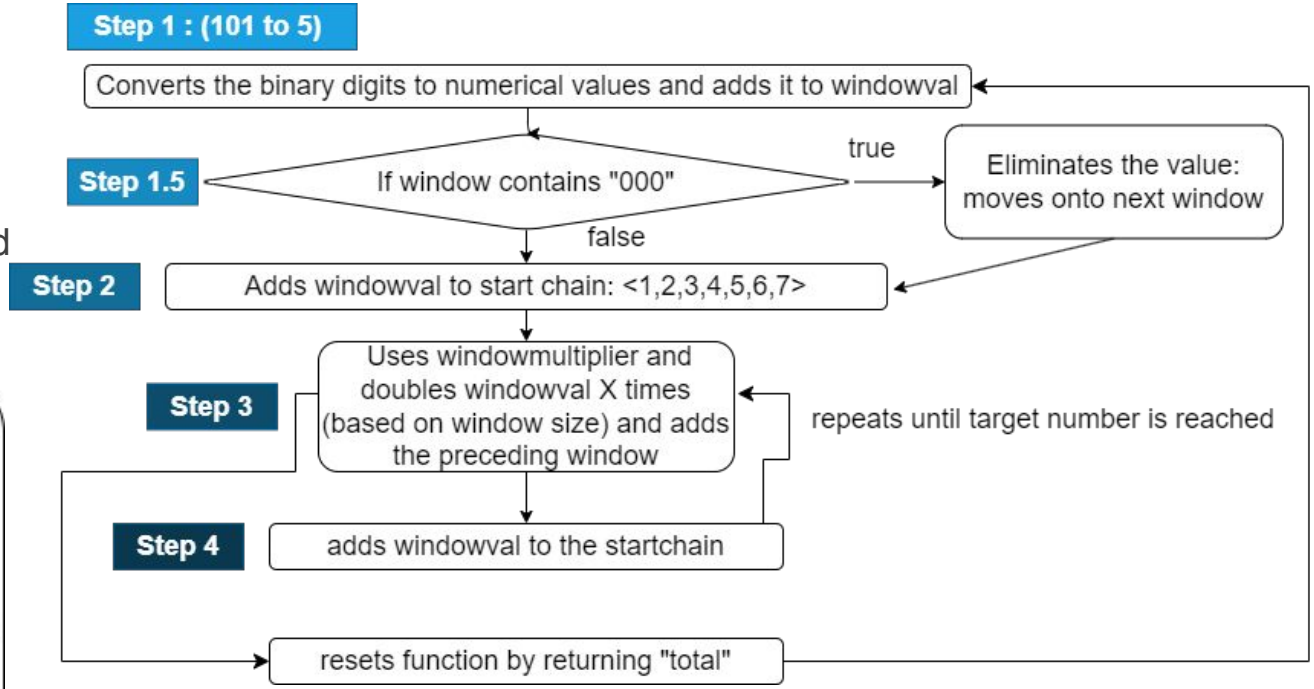
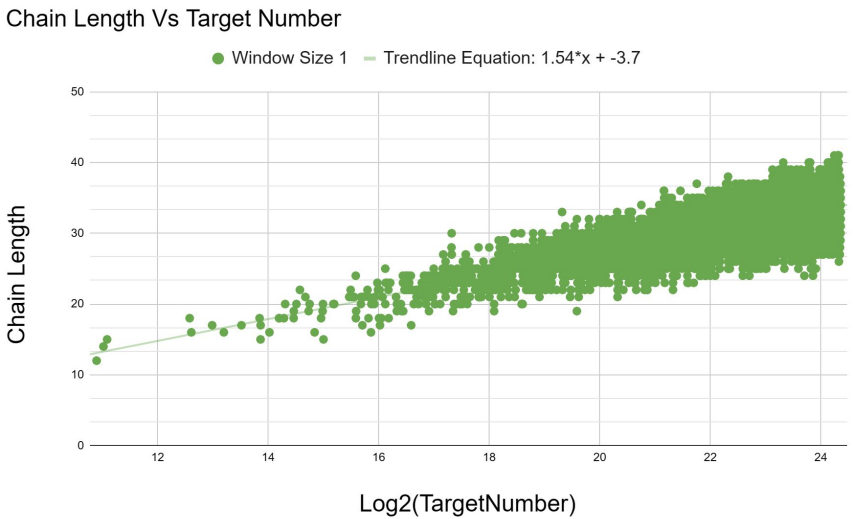


Figure 4 - A Flow Chart of the function GetDecimal in the Algorithm

Chain Length Vs Target Number



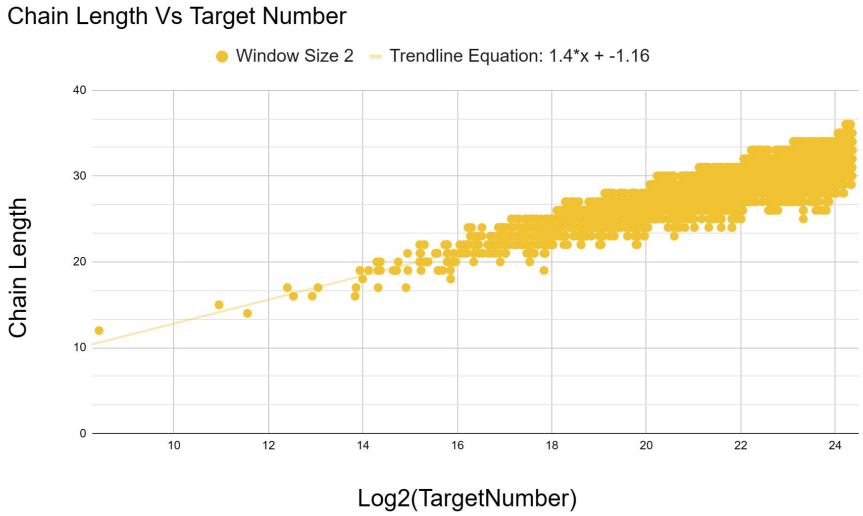
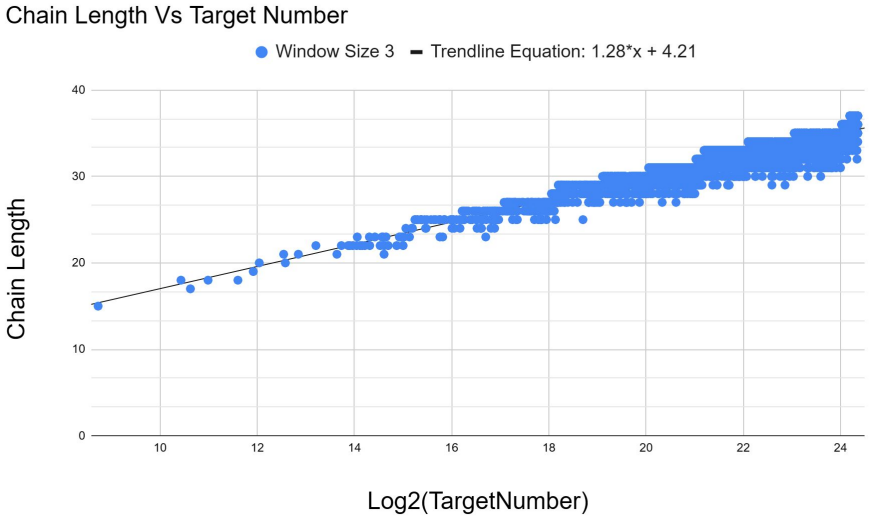


Figure 5 - “Graphs of Window Size 1 and 2 looking at Chain Length Vs Target Number log2(n). Comparing 20,000 unique target numbers (32-bit numbers) and their corresponding Chain Lengths. Using a thin line is the best fit line to approximate the relationship between Chain Length Vs Target Number. Processor used 5600x, 6 core and 12 threads @ 4.4 Ghz”

Chain Length Vs Target Number



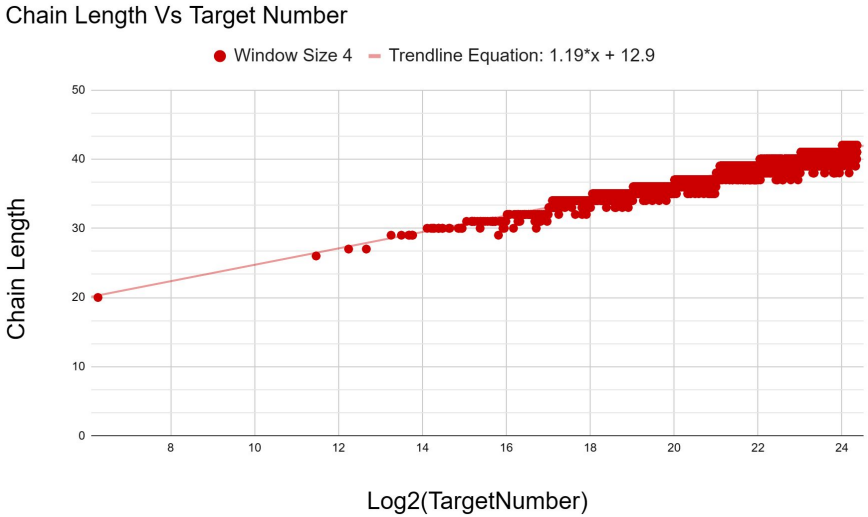


Figure 6 - “Graphs of Window Size 3 and 4 looking at Chain Length Vs Target Number log2(n). Comparing 20,000 unique target numbers (32-bit numbers) and their corresponding Chain Lengths. Using a thin line is the best fit line to approximate the relationship between Chain Length Vs Target Number. As the slope of the line decreases value, it seems to be more clumped together. Processor used 5600x, 6 core and 12 threads @ 4.4 Ghz”

Optimization of Window Method

- Residual value represents the difference between the y point graph and $\hat{x}$ the predicted value (on best fit line)
- Residual value looks how reliable are points
- Smaller residual values mean values are more precise to best fit line
- Window Size 4 is better for **bigger numbers up to 90+ bit numbers** however, window size 1 is **better for 1-18 bit numbers**

$$y = m\hat{x} + b$$
$$ChainLength - y = ResidualValue$$
$$\hat{x} = \log_2(TargetNumber)$$

Window Size	1	2	3	4
Max Residual Value	9.04	6.5	4.5	3.634
Mean Residual Value	1.81	1.11	0.77	0.487

Table 1 - A Table of Residual Values with Different Window Sizes

Window Size	Max Bit Size
1	**1-18**
2	**18-40**
3	**40-90**
4	**90+**

Table 2 - A Table of Predicted Ranges for Window Sizes

Implement Addition Chains onto the Elliptic Curve

- Normal ECC Implementation “path”: **1P, 2P, 3P, 4P, 5P**
  - **not fastest route** to reach 5P (P being the Point itself)

- Addition Chain Implementation “path”: **1P, 2P, 4P, 5P** it acts as a map for points on Elliptic Curve which can be calculated in Figure 7:
  - **provides a fastest route** for ECC to reach 5P

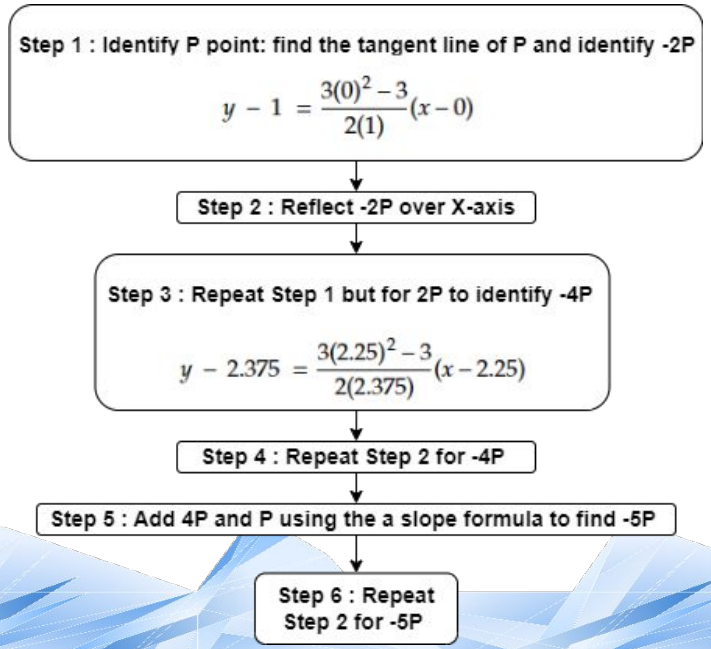


Figure 7 - Sample calculation for Chord and Tangent Process Diagram. This process doubles points (by finding the tangent line of a point. Ex: 2P to -4P) and adds points (by using the slope formula. Ex: 4P to -5P) on the elliptic curve.
The method “encrypts” points by doubling and adding points on the elliptic curve

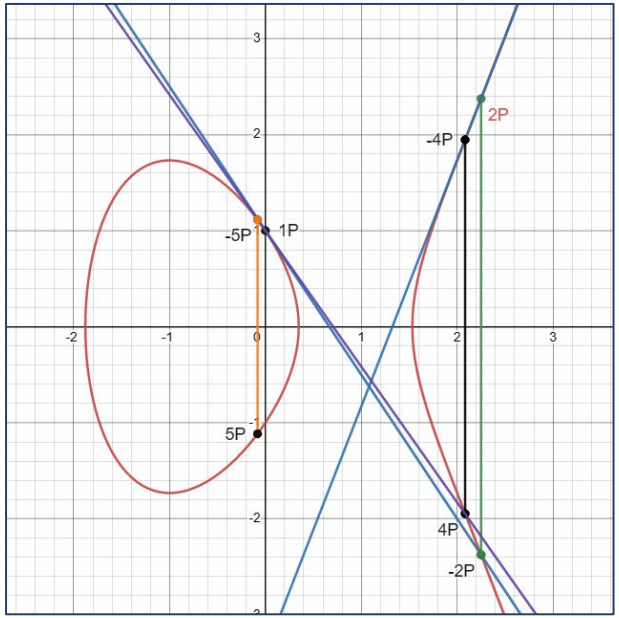


Figure 8 - “An elliptic curve with equation of y^2 = x^3-3x+1”

Real Life Applications

Example 1: Application to RSA Cryptography

- Faster exponential calculation
- In Figure 9, Encryption and Decryption are sped up because of the use of shortest addition chain

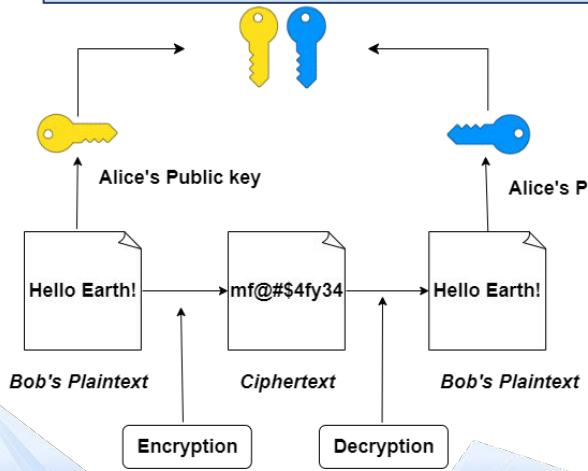


Figure 9 - A diagram of a common type of Cryptography: Public Encryption

Example 2: Other Cryptographic Primitives

- The use of digital signature algorithms, such as DSA, allows users to authenticate themselves to prove their identity in a given message.
- E-Commerce uses digital signatures extensively
- Using the shortest addition chain can increase encryption speed and secure digital signatures.
- Hackers will have a more challenging time spoofing the given digital signatures.

Conclusion

Overview

Created shortest addition chain algorithm via the window method approach.

- The lowest residual value showed that **Window Size 4** was the most reliable. It was projected that **Window Size 4 can handle larger numbers (90 bits+)** compared to Window Size 1 which is best for smaller numbers (1-18 bits).
- Feasibility study: This project is free to replicate and did not use any server or storage costs because of its low scale, but in a larger scale implementation, **server costs can increase.**
  - Even with increased storage costs, using the shortest addition chain **decreases memory usage and existing ECC's runtime**, saving money overall.

Avenues for Further Research

1: Looking at 256-bit numbers and see how they behave within the algorithm

2: Looking at eliminating terms in the start chain <1,2,3,4,5,6,7> to use less memory

3: Looking at different methods such as the run-length, Tunstall-code, hybrid, and other ways of generating shortest addition chains.

4: Exploring other chains such as addition subtraction chains and addition multiplication chains and how they can speed implementations of ECC.

Figure 10 - “Future Work Possible to Further Build Upon this Study”

References

- Diagrams & Graphs Credit to: The Author
- Liang, X., Li, Y., & Li, J. (2008). A novel shortest addition chains algorithm based on Euclid algorithm. In 2008 International Conference on Computer Science and Software Engineering (Vol. 5, pp. 1-4). IEEE. https://ieeexplore.ieee.org/document/4679046
- Li, Y., & Ma, Q. (2010). Design and implementation of layer extended shortest addition chains database for fast modular exponentiation in RSA. In 2010 International Conference on Web Information Systems and Mining (Vol. 1, pp. 136-139). IEEE. https://ieeexplore.ieee.org/document/5662298
- Kunihiro, N., & Yamamoto, H. (2000). New methods for generating short addition chains. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 83(1), 1-8. https://www.researchgate.net/publication/253718375_New_Methods_for_Generating_Short_Addition_Chains
- Hutchinson, A., & Karabina, K. (2019). Constructing multidimensional differential addition chains and their applications. Journal of Cryptographic Engineering, 9(1), 1-19
- Shand, M., & Vuillemin, J. (1993). Fast implementations of RSA cryptography. In Proceedings of 11th Symposium on Computer Arithmetic (pp. 252-259). IEEE
- Carlet, C., & Pott, A. (2010). Boolean functions for cryptography and error correcting codes. In C. Carlet & K. Feng (Eds.), Sequences and their applications - SETA 2010 (pp. 257-277). Springer1
- Bahig, H. M. (2006). Improved generation of minimal addition chains. Computing, 78(2), 161–172. https://doi.org/10.1007/s00607-006-0170-6