International Journal of Science and Research (IJSR)
ISSN: 2319-7064
Impact Factor 2024: 7.101

A Survey on Addition Chains Generation Methods

Anupama M[1], Dr. Shijo M Joseph[2], Dr. K. Mani[3]

[1]Research Scholar, Department of IT Kannur University

anupama.cs.mgc[at]gmail.com

[2]Professor, Department of Computer Science, Mahatma Gandhi College, Iritty

shijomjose71[at]gmail.com

[3]Assistant Professor, Department Artificial Intelligence, St. Joseph's College (Autonomous)

mani_ai2[at]mail.sjctni.edu

Abstract: ***As the usage of internet is increasing exponential, the sensitive/confidential data transmitted through it must be protected so that the attacker could not understand such sensitive/confidential data. Cryptography is a way of achieving data confidentiality. The study involves mathematical techniques designed to secure digital information prevent them from being collected and exploited by adversaries. Various cryptographic algorithms were being developed to secure data from security breach. In many public-key algorithms like RSA, encryption/decryption is of the form x^n mod p where x is plaintext/cipher text and n is encryption/decryption key, a positive integer. As n is too large, computing x^n more time which ultimately causes customers impatience and dissatisfaction. To reduce the encryption/decryption time, repeated multiplications are done, but it takes (n-1) repeated multiplications. To reduce the number of multiplications further, one way is the use of addition chain. Various methods have been developed to generate the addition chains. This paper reviews a variety of existing methods in the literature for generating optimal addition chains.***

Keywords: Cryptography, Public-key Cryptography, Addition Chain, Optimal Addition Chain

1. Introduction

A cryptosystem is a five –tuple $(\mathcal{P}, \mathcal{C}, \mathcal{K}, \mathcal{E}, \mathcal{D})$ where the following conditions are satisfied: $\mathcal{P}$ is a finite set of possible plaintext; $\mathcal{C}$ is a finite set of possible cipher text; $\mathcal{K}$, the key space, is a finite set of possible keys; For each $K \in \mathcal{K}$ there is an encryption rule $e_K \in \mathcal{E}$ and a corresponding decryption rule $d_K \in \mathcal{D}$. Each $e_K : \mathcal{P} \to \mathcal{C}$ and $d_K : \mathcal{C} \to \mathcal{P}$ are functions such as $d_K(e_K(x)) = x$ for every plaintext element $x \in \mathcal{P}$. [1]

Generally, cryptographic algorithms are divided into two types, private-key or symmetric-key cryptography (SKC) and public-key cryptography (PKC). In PKC, two different keys are used for encryption and decryption ; i.e., for every $k_e \in K$, there exists $k_d \in K'$ and $k_e \neq k_d$, where k_e and k_d are encryption and decryption key respectively [2].In PKC, the sender uses public-key of receiver to encrypt the plaintext (M or P)while the recipient uses his/her private-key to decrypt the ciphertext (C). A widely used technique used in PKC is the Modular or field exponentiation. RSA, ElGamal, ECC etc., are the most popular PKCs[3], [4], [5], [6]. Field exponentiation consists in finding a positive integer b satisfying the equation $b \equiv a^e \bmod p$, where a is a positive integer within the range [0, 1, 2,..., p-1], e is an arbitrary positive number and p is a prime number. This is a characteristic of exponentiation. In this, the exponent base a multiplied e times itself, and this problem is known as repeated multiplications (RMs). In general, to performa^e, (e-1) RMs are used. One elegant way of reducing the number of multiplications (Ms) is using addition chains (ACs).

Usually, ACs give a very easy way of computing x^e for given x and e. The optimal AC (OAC) for an integer e gives the least number of Ms needed to compute x^e. The length of an OAC for an integer e is usually denoted by l (e). Tables of OAC lengths have been used to benchmark of new algorithms. Positive integer addition on the shortest chain issues has been a lot of meaningful results [7], [8], [9]. However, most of them study the theoretical study of mathematics. Many researches and explorations concentrate on finding a short, not necessary minimal AC, while few papers study generating all OACs [10], [11].

To generate the ACs for an integer e, commonly used algorithms are deterministic and non-deterministic also called evolutionary algorithms. Factors method, sliding window method are examples of deterministic methods. Evolutionary Algorithms (EAs) are inspired by the idea of either natural evolution or social behavior of insects, birds, animals etc. Even though, they produce OACs, it is not possible to obtain them in a single run and also it is a consuming process. Genetic Algorithm (GA) Evolutionary Programming (EP), Particle Swarm Optimization (PSO), Simplified Swarm Optimization (SSO), Bacteria Foraging Optimization (BFO) etc. are some examples of EAs and concepts of said EAs are used to generate the OACs.

The rest of this paper is structured as follows. Various mathematical definitions of ACs are presented in section 2. A thorough literature regarding deterministic ways of generating OACs for an integer e is discussed in section 3. Non-deterministic ways of generating OACs are presented in section 4. Finally, section 5 ends with conclusion.

2. Mathematical Definitions of Addition Chain

This section presents the various mathematical definitions of ACs.

2.1 Definition (Addition Chain)

An addition chain for n is defined to be a sequence $(a_0, a_1 \ldots . a_r)$ such that $a_0 = 1, a_r = n$ and, for any $1 \leq k \leq r$, there exist $0 \leq i, j < k$ such that $a_k = a_i + a_j$ the number r is called the length of the addition chain. The shortest length among addition chains for n, called the addition chain length of n, is denoted $l(n)$ [12].

It is noted that if the value of *n*is relatively small, the exact value of l (*n*)is known. But, for large n, it is known thatit is very difficult to find.

The creation of each element of an AC is called a step. For an AC, $1 = a_0 \leq a_1 \leq \cdots \leq a_r = n$, the following steps are involved: Doubling step: $= 2a_{i-1}$, $i > 0$, Non-doubling step: $= a_j + a_k$, $i > j > k \geq 0$, The steps of the form $a_i = 2a_j$, $j \leq i - 2$ are defined as non-doubling steps. [13] big step and small step can be defined as $\lambda(a_i) = \lambda(a_{i-1}) + 1$ and $\lambda(a_i) = \lambda(a_{i-1})$ respectively. Thus, $l(n)$ can be split into two components as $l(n) = \lambda(n) + S(n)$. From the above, it is understood that the first step is always a doubling step. A doubling step is always a star step and never a small step. A doubling step must be followed by a star step. If step *i*is not a small step, then step $i + 1$ is either a small step or a star step or both. It is noted that not all doubling steps are big steps but big steps are always doubling. Finding OACs amounts to minimizing the number of small steps across all possible chains [13], [14].

2.2 Definition (Addition Chain)

An AC [14] is a finite sequence of positive integers called elements, $1 = a0 \leq a1 \leq a2 \leq \ldots \leq ar = e$ with the property that for all i>0 there exist a_j, a_k with $a\,i = aj + a\,k \; and \; r \geq i \geq j \geq k \geq 0.$

An optimal OAC is the one which has the shortest possible lengthand it is a strictly increasing sequence as duplicate chain elements could be removed to shorten the chain. It is noted that for the given integer e, many ACs are possible. But for finding at least one of the OACs is an NP-hard problem [15]

2.3 Definition (Addition-Subtraction Chain)

It is similar to an AC except that the coordinate $a_i = a_r + a_s$ is replaced by $a_i = a_r - a_s$. For example, AC for n=31 is1-2-4-8-10-20-30-31and its l (31)=7 whereas when Addition-Subtraction Chain (ASC) is used, it is 1-2-4-8-16-32-31 and its l (31)=6. In this way, for some e, ASC is used to reduce the length of OAC [16].

3. Literature Review

Literature Review of Deterministic Methods to Generate ACS

Bergeron et al. [17] introduced a unified method for computing short Addition Chains (ACs) using continued fraction expansions. Their framework encompasses various popular AC generation techniques, including the binary method and the factor method. They also recognized a common upper bound on the complexity of continued fraction approaches based on the selected approach. As a result, the total number of operations necessary to generate ACs for all integers up to n is O ($n\log^2 n\gamma_n$), where γ_n is the complexity of computing the set of choices corresponding to the approach.

Daniel M. Gordon[18] discussed various methods for fast exponentiation. He categorized the strengths and weaknesses of currently known techniques into window methods, special groups, and precomputation strategies.

R. Raveen et al. [19] proposed the Golden Ratio Addition-Reduction Chain (GRASC) method to generate efficient doubling-free short addition-reduction chains using the exact golden ratio. This method reduces the average chain length by 12%–28%, which reduces storage requirements and is suitable for small memory control devices implementing ECC. Experimental evaluation on 10, 000 randomly selected 160-bit integers shows that GRASC achieves a significantly shorter chain length of 258 compared to the Fibonacci-addition (358), signed Fibonacci-addition (322), windowed Fibonacci-addition (292), and efficient addition chain (EAC) methods (320).

Zulkarnain Md Ali et al.[20]developed an efficient computation algorithm for the addition-chain-based LUC cryptosystem, demonstrating a reduction in computation time compared to existing algorithms. This advancement enhances overall computational efficiency while also minimizing some of the iterations required in the LUC cryptosystem's calculations.

Neil Michael Clift [13] discusses the calculation of Optimal Addition Chains (OACs)in this paper. It presents a new algorithm that significantly outperforms the best-known methods when calculating sequences of OACs. Although the algorithm is slower for individual values compared to existing methods, it eliminates the need for tables of pre-calculated values, making it suitable for computing OACs for point values beyond the current chain limits. The lengths of all OACs for n≤ 232 were calculated, leading to the rejection of the hypothesis l (2n) ≥l (n)l. Additionally, the exact equality of the Scholz–Brauer conjecture, l (2n−1) =l (n)+n−1is verified for several new values.

Amadou Tall [21] proposed a generalization of Lucas Addition Chains (ACs) known as the Lucas addition-reduction chain (LASC). This method provides the minimum addition-reduction chains for an infinite set of integers and serves to demonstrate the optimality of Lucas ACs in various special cases. Similar to the binary method,

Lucas ACs yield minimum chains for all even integers with a Hamming weight of three. The Lucas addition-reduction chains produce shorter chains than traditional Lucas chains, suggesting that they may be particularly well-suited for elliptic curve cryptosystems (ECC) that utilize Lucas chains.

Ning Zhang and Shichong Tan [22] introduced a series of algorithms for elliptic curve scalar multiplication k[P] utilizing Fibonacci numbers. In their approach, k represents a scalar and P denotes a point on the elliptic curve. They developed a new computation for k[P] using a doubling-free addition chain. The authors analyzed the efficiency and security of this method, demonstrating through complex notation that it achieves both high efficiency and strong security.

Brian Koziel et al. [23] introduced a novel method for computing fast addition chains (ACs) leveraging isogeny primes. Their study focused on the characteristics of smooth isogeny primes, and they developed techniques to reduce the temporary register consumption associated with large exponentials, making their methods applicable to both software and hardware implementations of ACs. Additionally, they employed these procedures to compare various isogeny primes based on the complexity of the addition chains.

M. Hazem and Bahig [24] proposed an optimal parallel algorithm for small addition chains (ACs). This innovative algorithm is based on the right-left binary algorithm and utilizes a barrel shifter circuit. Designed for the EREW PRAM (Exclusive Read Exclusive Write Parallel Random-Access Machine) model, the algorithm achieves a time complexity of O ($(\log n)^2$)

Shuanggen Liu et al.[25] introduced a new Fibonacci-type sequence aimed at reducing the length of addition chains (ACs), which they refer to as a deformed Fibonacci-type AC (DFAC). This DFAC, built on a scalar multiplication algorithm, demonstrates an improvement over other scalar multiplication algorithms, achieving performance gains of 4% to 18%. Experimental results indicate that the DFAC method reduces the average chain length by 38% to 55% compared to other doubling-free AC methods. Notably, this algorithm does not exhibit any gaps when compared to other AC algorithms. Furthermore, the DFAC method consistently performs the operation 2P+Q2P + Q2P+Q in each step and is resistant to simple power attacks (SPA).

P. Anuradha Kameswari and B. Ravitheja [26] generated AC of length $3\lfloor \log n \rfloor - 1$ that yield the Lucas sequences V_n by using the formulas V_{2n} (a, 1), V_{2n+1} (a, 1) and $V_{2n}-1$, an AC of length $2\lfloor \log n \rfloor$ that yield the Lucas sequences V_n by using the formulas V_{2n} (a, 1) and V_{2n+1} (a, 1). Also generated Lucas AC of length $2\lfloor \log n \rfloor - 1$ yield the Lucas sequences V_n by using only one formula V_{x+y} (a, 1), for x, y such that x, y, x −y are in the Lucas AC. This study on ACs for Lucas sequences gives a cross-sectional view in understanding and evaluating similar computations like the point addition on ECs.

Shuang-Jen Liu and Hui Zhao [27] propose a novel fast and secure algorithm for arithmetic on elliptic curves (AC) using Pell Lucas sequences. Their approach introduces the Pell Lucas Type Chain (PLTC), which effectively combines mixed coordinates, thereby minimizing computational overhead. The energy curve of this new algorithm demonstrates uniformity, and the authors claim it is resistant to Side-Channel Attacks (SPA). Theoretical predictions and simulation experiments indicate that the k[P] based on the PLTC method achieves a speed increase of 22.7% compared to the traditional golden ratio algorithm.

Christophe Negre and Thomas Plantard [28] propose that multiplicative division can enhance the performance of the RSA algorithm, independent of the arithmetic context. In their paper, they focus on efficient RSA modular exponentiation techniques designed to mitigate simple side-channel attacks, such as timing attacks and simple power analysis. To achieve this, they divide the integer x mod n into two half-size integers, leveraging this division to modify the square-multiplicative approach. This modification transforms the computation into a regular exponentiation sequence by consistently multiplying by a half-size integer. Their methods demonstrate a reduction of approximately 16% in word operations compared to the Montgomery ladder, square-always, and square-multiplicative-always exponentiation methods.

Hazem M. Bahig and Yasser Kotb [29] introduced a novel algorithm in their paper for determining the minimum length addition chain (AC) of an integer n. Their research on multicore systems exposed the running time of this new algorithm exceeds that of traditional sequential algorithms. Furthermore, the maximum speed attained by the proposed algorithm is 2.5 times superior than that of the famous sequential algorithm.

Hazem M. Bahig et al. [30] proposed a new evolutionary algorithm (EA) called EAC, which enhances traditional EA components-specifically representation, population, mutation, and survivor selection-in a straightforward and efficient manner. They represent the population as a 2D array, where the first dimension corresponds to the number of individuals and the second dimension represents the variable length of each individual. A fitness function is employed to evaluate individuals against a target solution. The authors modified the initial population strategy by assessing the difference between a target number eee and the last generated element. To create mutated chains, they utilized two auxiliary arrays: Aux1, which retains the best mutation from eee, and Aux2, which generates the mutated chain from a previous chain Ek. At the conclusion of k iterations, the best mutated version of AC for e is chosen as the offspring. Their survivor selection algorithm simplifies the process by eliminating the fitness calculation for q randomly selected chains and also removes duplicate chains. The EAC algorithm was implemented and tested against three other algorithms, demonstrating superior performance when applied to natural numbers represented with 10, 11, 12, 13, and 14 bits.

In their paper, Shuang-Gen Liu et al.[31] Introduced a practice to improve the performance of k[P] through a better type of addition chain (AC). They present an algorithm that utilize a trade-off among multiplication and squaring in Jacobian coordinates to directly compute 5P, effectively reducing the computational cost from 15M+10S to 8M+16S, where S and M denote squaring and multiplication operations in a finite field, respectively. In addition, they propose an algorithm that integrates z-point addition with the properties of the generalized Fibonacci sequence to compute 5P+Q. This enhanced AC is termed Fibonacci Type AC (IFTAC), which, for a key length of 160 bits, results in a chain length of 65, thereby significantly shortening the chain length compared to previous algorithms. Subsequently they propose k[P].EC, which leverages the properties of IFTAC to mitigate side-channel attacks (SPA), since the computation of 5P+Q is consistently repeated in each iteration.

Edward G. Thurber and Neil M. Clift [32] establish a remarkable connection between alternative computations (ACs) and vector chains, such that the search for ACs can be formulated as a depth-first search tree. The required search limits are significantly reduced, simplifying the process.

Yuanchao Ding et al.[33] proposed an innovative technique for generating short alternative computations (ACs). A simple power-tree method can significantly reduce the time complexity of ACs by simplifying the power tree even if the length is increased. To improve efficiency, a cross-window method (CWM) and its variant are introduced, which improve the traditional window method by using cross-correlation for window management and pre-computation. The proposed approach produces shorter ACs, thereby achieving a 9.5% reduction in AC length.

Lasheras, Ana, et al [34] proposed a residual check-based lightweight technique to protect circuits applying the RSA algorithm against DFA and HWT at runtime. They claim that the proposed algorithm gives 100% result against fault tolerance and HWT detection among all RSA algorithms in the Trust-Hub benchmark suite. This proposed system has no effect on the operating frequency, as it uses only a maximum area increase of 3% and a dynamic power consumption of about 18%.

Sakkari, Deepak S., and Mohammed Mujeer Ulla[35], discussed various topics such as the invariant logarithm problem and the Diaphantus method for identifying points on an elliptic curve, they conclude that high efficiency and small key sizes give elliptic curve cryptography an advantage over its peer cryptographic techniques.

Reddy, Sathi Sarveswara, et al[36] implemented an RSA and Pailler using PSO-based hardware/software co-design on Xilinx Virtex-7 FPGAs which saves resources as follows: RSA encryption 60.7% (area) and 65.3% (DSP);Paillier encryption - 46.3% (Area) and 40% (DSP); RSA Decryption 60.7% (Area) and 65.3% (DSP); Pillar Decryption - 73.7% (Area) and 66.6% (DSP). In addition, they got 1024-bit area-time improvements as RSA encryption - 2.7x; Pillar Encryption - 2x; RSA Decryption - 2.7x, Pillar Decryption - 4.6x.

Mohamed, Mohamad Afendee, et al [37] developed the concept of an addition chain that can enhance the performance of ECC. By considering the selected curve, the formulation to express the original equation, the operation process and the arithmetic choices, they proposed a two-module method called the chain generator based on heuristics method and an integer recoding method.

Literature Review of Non-Deterministic Methods to Generate ACS

Nareli Cruz-Cortes et al. [3] introduced an artificial immune system (AIS) heuristic aimed at develop shorter alternative computations (ACs) in their paper. They employed the AIS to identify the shortest ACs for exponents of moderate size (less than 20 bits) and for larger exponents typically used in cryptographic applications (ranging from 128 to 2048 bits). The algorithm was developed based on the clonal selection principle and incorporated key components such as antibody construction, a hyper mutation operator, an immune memory mechanism, and the clonal selection algorithm. The proposed AIS heuristic demonstrated an impressive capability to discover nearly all optimal ACs for fixed exponents with $e < 4096e$, achieving a high success rate of 99.6%.

Arindam Sarkar and J.K. Mandal [38] applied a swarm algorithm to enhance the optimization potential of an algorithm using soft computing tools. They proposed a swarm intelligence-based Faster Public-Key Cryptography in Wireless Communication (SIFPKC). This algorithm begins with an initial population of valid and complete particles. They utilize operators such as the local best and global best (Gbest) positions to generate potential valid particles through velocity update rules. Additionally, they compare the best solution achieved by SIFPKC with several existing techniques.

Eduardo Vázquez-Fernández et al. [39] introduced a genetic algorithm (GA) featuring a mutation mechanism that leverages both Gaussian and uniform distributions to reduce alternative computations (ACs) for small exponents. In their study, they integrated these distributions into the mutation mechanism to enhance the GA's performance. The proposed method incorporates a mutation operator that utilizes uniform and/or Gaussian distributions, thereby optimizing the effectiveness of the genetic algorithm.

In their paper, K. Mani and M. Vishwambari [40] propose two innovative graph-based methods for generating optimal alternative computations (OACs) for an integer e. In these methods, the vertices of the graph stand for the numbers in the AC, while the edges indicate the transitions from one number to another. Method 1, termed GBAPAC, generates all possible OACs by considering the edge weights associated with all potential numbers derived from each number in the AC. Method 2, named GBMAC,

focuses on generating the minimum OACs by addressing conflicting edges that originate from each number. Additionally, the OACs produced by these methods are evaluated against existing assumptions in the literature regarding ACs.

Mauricio Olguín-Carbajal and others [41] conducted a study on generating minimum alternative computations (ACs) using evolutionary strategies. They developed a minimum length AC generator that requires fewer calls to the objective function compared to other approaches, such as bio-inspired algorithms like particle swarm optimization (PSO) and genetic algorithms (GA). By minimizing the number of objective function calls, their method reduces computational effort and generation time, resulting in improved computational cost while still achieving competitive results. The study highlights that evolutionary strategies (ESs) can optimize time relative to other algorithms due to their rapid convergence. However, this fast convergence can diminish the algorithm's ability to explore the solution space, potentially impacting the quality of the proposed solutions. Overall, evolutionary algorithms (EAs) present a viable technique for creating short ACs.

A. Mullai and K. Mani [42] proposed algorithms for generating alternative computations (ACs) aimed at increasing the speed of encryption and decryption processes while enhancing the security of public key cryptosystems (PKCs) such as RSA and ECC. The ACs are generated using evolutionary algorithms, including particle swarm optimization (PSO), simplified swarm optimization (SSO), and bacterial foraging optimization (BFO). Once the optimal alternative computations (OACs) are generated, they are integrated into the RSA and ECC encryption and decryption processes using Android and Windows emulators. The analysis of processing time, power consumption during encryption and decryption, and the overall security of these algorithms reveals that the performance of the PKCs is significantly improved.

K. Mani and A. Mulla [43] introduced an alternative computation (AC) using the Bacterial Foraging Optimization Algorithm (BFOA), which is inspired by the foraging behavior of bacteria. Their proposed algorithm achieved an AC with an optimal length nearly equivalent to that of existing ACs for certain challenging exponents. In this approach, a virtual bacterium serves as the search agent, similar to other evolutionary algorithms such as Artificial Immune Systems (AIS), Genetic Algorithms (GA), and Evolutionary Programming (EP), with the cost or fitness function calculated as l (e)l (e)l (e). Notably, the authors did not incorporate a swarming step for AC generation. Experimental results indicated that the BFOA produced the same optimal length of AC for integers up to 1024, closely matching the performance of other evolutionary algorithms like AIS, GA, and EP.

4. Conclusion

This paper presented and explored some AC techniques to find the OACs. Based on the literature survey, it is concluded that various techniques have been used to generate AC research during the last few years and soon so many drastic changes and steps would be generated during the coming year. Overall, this paper is intended towards the creation of a platform for similar researchers involved in the AC generation field based on which they can choose, devise new technologies and also improve upon already existing techniques. This survey paper will always be useful for the new researchers to reduce the encryption/decryption time of public-key cryptosystems like RSA, ElGamal, ECC etc., by incorporating the AC into encryption/decryption operations.

References

[1] D. R. Stinson and M. Paterson, Cryptography: Theory and Practice. CRC Press, 2018.
[2] H. Delfs and H. Knebl, Introduction to Cryptography: Principles and Applications. in Information Security and Cryptography. Berlin, Heidelberg: Springer, 2007. doi: 10.1007/3-540-49244-5.
[3] N. Cruz-Cortes, F. Rodriguez-Henriquez, and C. A. Coello Coello, “An Artificial Immune System Heuristic for Generating Short Addition Chains”, IEEE Trans. Evol. Comput., vol. 12, no. 1, pp. 1–24, Feb. 2008, doi: 10.1109/TEVC.2007.906082.
[4] N. Cruz-Cortés, F. Rodríguez-Henríquez, R. Juárez-Morales, and C. A. Coello Coello, “Finding Optimal Addition Chains Using a Genetic Algorithm Approach”, in Computational Intelligence and Security, vol. 3801, Y. Hao, J. Liu, Y. Wang, Y. Cheung, H. Yin, L. Jiao, J. Ma, and Y.-C. Jiao, Eds., in Lecture Notes in Computer Science, vol. 3801, Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 208–215. doi: 10.1007/11596448_30.
[5] A. León-Javier, N. Cruz-Cortés, M. A. Moreno-Armendáriz, and S. Orantes-Jiménez, “Finding Minimal Addition Chains with a Particle Swarm Optimization Algorithm”, in MICAI 2009: Advances in Artificial Intelligence, vol. 5845, A. H. Aguirre, R. M. Borja, and C. A. R. Garciá, Eds., in Lecture Notes in Computer Science, vol. 5845, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 680–691. doi: 10.1007/978-3-642-05258-3_60.
[6] D. Mahto and D. K. Yadav, “RSA and ECC: A Comparative Analysis”, vol. 12, no. 19, 2017.
[7] D. Dobkin and R. J. Lipton, “Addition Chain Methods for the Evaluation of Specific Polynomials”, SIAM J. Comput., vol. 9, no. 1, pp. 121–125, Feb. 1980, doi: 10.1137/0209011.
[8] F. Bergeron, J. Berstel, S. Brlek, and C. Duboc, “Addition chains using continued fractions”, J. Algorithms, vol. 10, no. 3, pp. 403–412, Sep. 1989, doi: 10.1016/0196-6774 (89)90036-9.
[9] H. M. Bahig and H. M. Bahig, “A new strategy for generating shortest addition sequences”, Computing, vol. 91, no. 3, pp. 285–306, Mar. 2011, doi: 10.1007/s00607-010-0119-7.
[10] W. Jiang, J. Zhang, and J. Li, “A Multi-agent Supply Chain Information Coordination Mode Based on Cloud Computing”, TELKOMNIKA Indones. J. Electr. Eng., vol. 11, no. 11, pp. 6427–6433, Nov. 2013, doi: 10.11591/telkomnika.v11i11.3453.

[11] P. Downey, B. Leong, and R. Sethi, "Computing Sequences with Addition Chains", SIAM J. Comput., vol. 10, no. 3, pp. 638–646, Aug. 1981, doi: 10.1137/0210047.
[12] H. Altman, "Internal structure of addition chains: Well-ordering", Theor. Comput. Sci., vol. 721, pp. 54–69, Apr. 2018, doi: 10.1016/j.tcs.2017.12.002.
[13] N. M. Clift, "Calculating optimal addition chains", Computing, vol. 91, no. 3, pp. 265–284, Mar. 2011, doi: 10.1007/s00607-010-0118-8.
[14] M. K. Mani, "Generation of Addition Chain using Deterministic Division Based Method", Eng. Technol.
[15] E. G. Thurber, "Addition chains - an erratic sequence", Discrete Math., vol. 122, no. 1–3, pp. 287–305, Nov. 1993, doi: 10.1016/0012-365X (93)90303-B.
[16] "The Art of Computer Programming." Accessed: Mar. 17, 2025. [Online]. Available: https://www-cs-faculty.stanford.edu/~knuth/taocp.html
[17] F. Bergeron, J. Berstel, and S. Brlek, "Efficient computation of addition chains", J. Théorie Nr. Bordx., vol. 6, no. 1, pp. 21–38, 1994, doi: 10.5802/jtnb.104.
[18] D. M. Gordon, "A Survey of Fast Exponentiation Methods", J. Algorithms, vol. 27, no. 1, pp. 129–146, Apr. 1998, doi: 10.1006/jagm.1997.0913.
[19] R. R. Goundar, K. Shiota, and M. Toyonaga, "New Strategy for Doubling-Free Short Addition-Subtraction Chain".
[20] Z. M. Ali, M. Othman, M. R. Muhd, and M. N. Sulaiman, "Computation of Cryptosystem based on Lucas Functions using Addition Chain", in 2010 International Symposium on Information Technology, Jun. 2010, pp. 1082–1086. doi: 10.1109/ITSIM.2010.5561514.
[21] A. TALL, "A generalization of the Lucas addition chains", 2011, 2011/378. Accessed: Mar. 17, 2025. [Online]. Available: https://eprint.iacr.org/2011/378
[22] N. Zhang and S. Tan, "Elliptic Curve Scalar Multiplication Based on Fibonacci Number", in Proceedings of the 2013 5th International Conference on Intelligent Networking and Collaborative Systems, in INCOS '13. USA: IEEE Computer Society, Sep. 2013, pp. 507–510. doi: 10.1109/INCoS.2013.157.
[23] B. Koziel, R. Azarderakhsh, D. Jao, and M. Mozaffari-Kermani, "On Fast Calculation of Addition Chains for Isogeny-Based Cryptography", in Information Security and Cryptology, K. Chen, D. Lin, and M. Yung, Eds., Cham: Springer International Publishing, 2017, pp. 323–342. doi: 10.1007/978-3-319-54705-3_20.
[24] H. M. Bahig, "A fast optimal parallel algorithm for a short addition chain", J. Supercomput., vol. 74, no. 1, pp. 324–333, Jan. 2018, doi: 10.1007/s11227-017-2129-0.
[25] S. Liu, G. Qi, and X. A. Wang, "Fast elliptic curve algorithm using deformed Fibonacci-type series", Int. J. Embed. Syst., vol. 10, no. 2, p. 104, 2018, doi: 10.1504/IJES.2018.090563.
[26] P. A. Kameswari and B. Ravitheja, "ADDITION CHAIN FOR LUCAS SEQUENCES WITH FAST COMPUTATION METHOD", vol. 13, no. 11, 2018.
[27] S.-G. Liu and H. Zhao, "SPA Resistant Scalar Multiplication Using Pell Lucas Type Chain", Int. J. Netw. Secur., vol. 21, no. 4, pp. 627–634, Jul. 2019, doi: 10.6633/IJNS.201907 21 (4).12.
[28] C. Negre and T. Plantard, "Efficient regular modular exponentiation using multiplicative half-size splitting", J. Cryptogr. Eng., vol. 7, no. 3, pp. 245–253, Sep. 2017, doi: 10.1007/s13389-016-0134-5.
[29] H. Bahig and Y. Kotb, "An Efficient Multicore Algorithm for Minimal Length Addition Chains", Computers, vol. 8, no. 1, p. 23, Mar. 2019, doi: 10.3390/computers8010023.
[30] H. M. Bahig, K. A., M. A., A. AlGhadhban, and H. M., "An Evolutionary Algorithm for Short Addition Chains", Int. J. Adv. Comput. Sci. Appl., vol. 11, no. 12, 2020, doi: 10.14569/IJACSA.2020.0111258.
[31] S.-G. Liu, X. Wang, Y.-W. Liu, and D.-J. Li, "Fast Scalar Multiplication Algorithms Based on 5P+Q of Elliptic Curve over GF (3m)", Int. J. Netw. Secur., vol. 23, no. 4, pp. 604–611, Jul. 2021, doi: 10.6633/IJNS.202107 23 (4).06.
[32] E. G. Thurber and N. M. Clift, "Addition chains, vector chains, and efficient computation", Discrete Math., vol. 344, no. 2, p. 112200, Feb. 2021, doi: 10.1016/j.disc.2020.112200.
[33] Y. Ding, H. Guo, Y. Guan, H. Song, X. Zhang, and J. Liu, "Some New Methods to Generate Short Addition Chains", IACR Trans. Cryptogr. Hardw. Embed. Syst., pp. 270–285, Mar. 2023, doi: 10.46586/tches.v2023.i2.270-285.
[34] A. Lasheras, R. Canal, E. Rodríguez, and L. Cassano, "Securing RSA hardware accelerators through residue checking", Microelectron. Reliab., vol. 116, p. 114021, Jan. 2021, doi: 10.1016/j.microrel.2020.114021.
[35] D. S. Sakkari and M. M. Ulla, "Review on Insight into Elliptic Curve Cryptography", in Modern Approaches in Machine Learning & Cognitive Science: A Walkthrough, vol. 1027, V. K. Gunjan and J. M. Zurada, Eds., in Studies in Computational Intelligence, vol. 1027, Cham: Springer International Publishing, 2022, pp. 81–93. doi: 10.1007/978-3-030-96634-8_8.
[36] S. S. Reddy, S. Sinha, and W. Zhang, "Design and Analysis of RSA and Paillier Homomorphic Cryptosystems Using PSO-Based Evolutionary Computation", IEEE Trans. Comput., vol. 72, no. 7, pp. 1886–1900, Jul. 2023, doi: 10.1109/TC.2023.3234213.
[37] M. A. Mohamed et al., "Addition chain heuristics in application to elliptic curve cryptosystems", Int. J. Adv. Appl. Sci., vol. 13, no. 3, p. 546, Sep. 2024, doi: 10.11591/ijaas.v13.i3.pp546-555.
[38] A. Sarkar and J. K. Mandal, "Swarm Intelligence based Faster Public- Key Cryptography in Wireless Communication (SIFPKC)", Int. J. Comput. Sci. Eng. Technol. IJCSET, vol. 3, no. 7, pp. 267–273.
[39] E. Vázquez-Fernández, C. Cadena, and D. A. Reyes-Gómez, "A genetic algorithm with a mutation mechanism based on a Gaussian and uniform distribution to minimize addition chains for small exponents", in 2016 IEEE Congress on Evolutionary

Computation (CEC), Jul. 2016, pp. 935–940. doi: 10.1109/CEC.2016.7743890.

[40] Nehru Memorial College, Puthanampatti, Trichy, TamilNadu, India-621 007, K. Mani, and M. Viswambari, "A New Method of Generating Optimal Addition Chain Based on Graph", Int. J. Math. Sci. Comput., vol. 3, no. 2, pp. 37–54, Apr. 2017, doi: 10.5815/ijmsc.2017.02.04.

[41] M. Olguin Carbajal, J. C. Herrera-Lozada, I. Rivera-Zárate, J. F. Serrano-Talamantes, R. Cadena-Martínez, and J. I. Vásquez-Gómez, "Minimum Addition Chains Generation Using Evolutionary Strategies", Comput. Sist., vol. 22, no. 4, Dec. 2018, doi: 10.13053/cys-22-4-2751.

[42] A. Mullai and K. Mani, "Enhancing the security in RSA and elliptic curve cryptography based on addition chain using simplified Swarm Optimization and Particle Swarm Optimization for mobile devices", Int. J. Inf. Technol., vol. 13, no. 2, pp. 551–564, Apr. 2021, doi: 10.1007/s41870-019-00413-8.

[43] M. K and M. A, "Generation of Addition Chain using Bacteria Foraging Optimization Algorithm", Int. J. Eng. Trends Technol., vol. 69, no. 2, pp. 32–38, Feb. 2021, doi: 10.14445/22315381/IJETT-V69I2P205.