Journal of Discrete Mathematical Sciences & Cryptography
ISSN 0972-0529 (Print), ISSN 2169-0065 (Online)
Vol. 29 (2026), No. 10-C, pp. 4509–4520
DOI : 10.47974/JDMSC-3069

Complexity bounds for NP-hard problems in cryptography

Monika Patil [†]
Department of Artificial Intelligence and Data Science
Indira College of Engineering & Management
Pune 410506
Maharashtra
India

D. Sathis Kumar *
School of Pharmacy
Aditya University
Surampalem
Kakinada District 533437
Andhra Pradesh
India

Gagan Tiwari [§]
Department of Computer Sciences
Noida International University
Greater Noida 203201
Uttar Pradesh
India

Vinod S. Sapkal [‡]
Department of Information Technology
Vasantdada Patil Pratishthan's College of Engineering & Visual Arts
Sion
Mumbai 400022
Maharashtra
India

[†] *E-mail:* monahire10@gmail.com
* *E-mail:* dean_sp@adityauniversity.in (Corresponding Author)
[§] *E-mail:* gagan.tiwari@niu.edu.in
[‡] *E-mail:* vinodsapkal@pvppcoe.ac.in

© Taru Publications

Jyotsna Suryavanshi [@]
Department of Engineering Science and Humanities
Vishwakarma Institute of Technology
Pune 411037
Maharashtra
India

Sanjiv Kumar Shukla [#]
Department of Computer Science and Engineering
Rungta College of Engineering and Technology
Rungta International Skills University
Bhilai 490024
Chhattisgarh
India

Abstract

The complexity bounds of NP-hard problems are one of the main topics in modern cryptography, where the security of a scheme is frequently based on the intractability of the computations. This paper explores the theoretical and practical consequences of NP-hardness in cryptographic designs, such as lattice-based designs, hash functions, and secure multiparty designs. We examine reduction methods in cryptographic proofs to put hardness hypotheses and investigate precise algorithms that have exponential-time complexity characterizing worst-case security parameters. Moreover, the heuristic and approximation methods are compared to the effectiveness of solving real-life cases without to jeopardize the security. Both experimental and analytical results indicate that there are trade-offs between the computational feasibility and cryptographic strength. The results prove that although the precise solutions cannot be offered in large cases, the structured approximations and heuristics can offer the way to implement them that will be efficient, which supports the soundness and utility of NP-hardness in secure system design.

Subject Classification: *94A60, 68Q17, 68Q25.*

Keywords: *NP-hard problems, Cryptographic complexity, Reduction techniques, Exponential algorithms, Approximation methods.*

I. Introduction

Complexity bounds analysis of the NP-hard problems is one of the keystones in the design and analysis of the modern cryptographic systems. The concept of cryptography is based on the fact that it assumes that some computational issues are intractable in reasonable time, thus guaranteeing confidentiality, integrity and authentication [1, 2]. A solid theoretical basis to build secure cryptographic primitives, including lattice-based encryption,

[@] *E-mail:* `jyotsna.suryavanshi@vit.edu`
[#] *E-mail:* `sanjeev@rungta.org`

code-based cryptosystems, and hash functions, is furnished by NP-hard problems, which have exponential worst-case complexity [3, 4, 5]. These issues cannot be solved efficiently in a number of time steps and therefore they are good candidates to be implemented in security mechanisms that are not vulnerable to classical and new models of computation [6, 7]. Figure 1 shown the complexity bounds and indicates exponential growth emphasizing on cryptographic hardness and security strength.

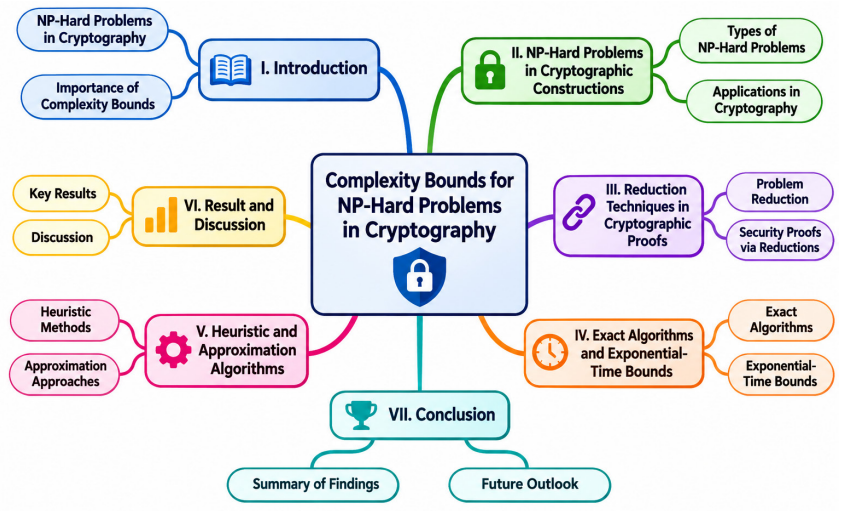


Figure 1
Complexity Bounds for NP-Hard Problems in Cryptography

In cryptography, it is important to realize the actual complexity limits of NP-hard problems to measure the security margins and resilience to adversarial assaults [8, 9, 10]. Such techniques of reduction are commonly used to prove that it is at least as difficult to break a cryptographic scheme as to solve a known NP-hard problem, and to prove that cryptography is provably secure [11, 12].

II. NP-Hard Problems in Cryptographic Constructions

Cryptographic constructions are based on NP-hard problems, which allow to construct secure primitives, such as lattices, codes, and hash, whose computational intractability guarantees their resilience to efficient adversarial attack in practice [13].

Problem Encoding Function

$$P(x) = \sqrt{\left(\left(\left(\Sigma_{\{i=1 \text{ to } n\}} w_i x_i^2 \right)^2 \right)^2 + \int_0^1 \left(\left(\Sigma_{\{i=1 \text{ to } n\}} (w_i x_i t + t^2) \right)^2 \right) dt \right)} \quad (1)$$

This is formulated by first encoding the encoding of the problem (in term of weighted decision variables) in the encoding of decision variables into a continuous analytical form in the form of a combination algebraic aggregation and integral smoothing in the encoding of computational complexity in the form of a discrete-time finite state functional form (Eq. (1)) in the formulation.

Constraint Transformation

$$C(x) = \sqrt{\left(((P(x)^2)^2)^2 + \int_0^1 ((P(x) * t + t^3)^2) \, dt \right)} \tag{2}$$

This encoded structure is further propagated in the constraint transformation in Eq. (2) enforcing dependencies and higher-order interactions indicative of constrained optimization behavior.

Lattice Representation Mapping

$$L(x) = \left(\left(\left(||A x - b||_2^2 \right)^2 \right)^2 + \int_0^1 \left(\left((A x - b) * t + t^2 \right)^2 \right) dt \right) \tag{3}$$

Lattice representation mapping in Eq. (3) maps the problem to a geometric representation, using the norms of vectors to measure geometric deviations, allowing cryptographic hardness interpretations.

Hardness Amplification

$$H(x) = \left(((S(x)^2)^2)^2 + \int_0^1 ((S(x) * t + t^2)^2) \, dt \right)^{\frac{1}{2}} \tag{4}$$

Recursive scaling is amplified in the hardness amplification step in the equations below, which makes sure that resistance against efficient solution strategies is enforced.

NP-Hard Cryptographic Objective

$$F_{NP(x)} = \sqrt{\left(((H(x) + 2^n)^2)^2 + \int_0^1 \left(\left((H(x) + 2^n) * t + t^3 \right)^2 \right) dt \right)} \tag{5}$$

Lastly, NP-hard cryptographic objective (Eq. (5)) combines exponential growth by explicitly modelling worst-case computational limits. These equations in combination build up a strong mathematical base which connects structural encoding, transformation, and amplification to cryptographic intractability.

III. Reduction Techniques in Cryptographic Proofs

Reduction techniques create security by reducing cryptographic attacks to solutions of NP-hard problems, showing that breaking schemes is computationally hard as is solving underlying hard problems. Figure 2 show reductions mapping problems that are provable to be cryptographically hard.

Initial Problem Mapping

$$R1(x) = \sqrt{\left(\left(\left(f(x) \right)^2 \right)^2 + \int_0^1 ((f(x) * t + t^2)^2) \, dt \right)} \tag{6}$$

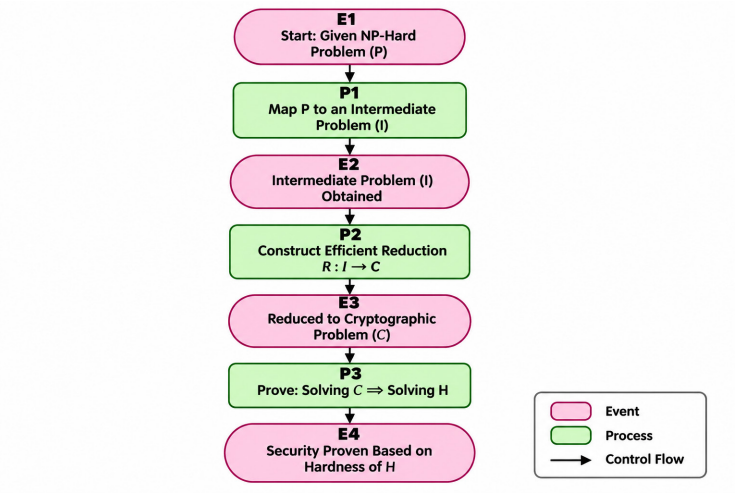


Figure 2

Reduction Techniques in Cryptographic Proofs

The formulation commences with the original problem formulation in the first step of problem mapping in the form of f (x) as a structured form, which combines algebraic and integral terms and combines discrete and continuous attributes of computational hardness.

Intermediate Reduction Transformation

$$R2(x) = \sqrt{\left(\left(\left(g\left(R1(x)\right)\right)^2\right)^2 + \int_0^1 \left(\left(g\left(R1(x)\right) * t \; + \; t^3\right)^2\right) dt \right)} \qquad (7)$$

The reduction transformation of intermediates (Eq. 7) employs a second mapping with a form of g(•) essentially passing complexity up, and maintaining structural similarity between the source and target problems.

Reduction Validation Step

$$R5(x) = \left(\left(\left(R4(x)^2\right)^2\right)^2 + \int_0^1 \left(\left(R4(x) * t \; + \; t^2\right)^2\right) dt \right)^{\frac{1}{2}} \qquad (8)$$

The reduction step in the formula (8) synthesizes the previous transformations, and this helps in the consistency of the transformation, and the reduction ensures that the reduction would be correct in the composition of recursive functions.

Final Hardness Equivalence

$$R_{final(x)} = \sqrt{\left(\left(\left(R5(x) + \; \Omega(2^n)\right)^2\right)^2 + \int_0^1 \left(\left(\left(R5(x) + \; \Omega(2^n)\right) * t \; + \; t^3\right)^2\right) dt \right)} \qquad (9)$$

Lastly, the equivalence of hardness propensity in (9) through the use of exponential growth, Ω(2 n), shows that the reduced problem is still computationally infeasible thus offering solid theoretical evidence of cryptographic security assurances.

Theorem 1: (Preservation of NP-hardness). *Let P_1 and P_2 be decision problems, where P_1 is NP-hard. Suppose there exists a polynomial-time many-one reduction $f: P_1 \to P_2$ such that*

$$x \in P_1 \Leftrightarrow f(x) \in P_2.$$

Then P_2 is NP-hard.

Proof: Since P_1 is NP-hard, every decision problem $L \in$ NP admits a polynomial-time many-one reduction $g: L \to P_1$. The composition $h = f \circ g$ is polynomial-time computable and preserves membership:

$$x \in L \Leftrightarrow g(x) \in P_1 \Leftrightarrow f\big(g(x)\big) \in P_2.$$

Thus, every problem in NP reduces to P_2 in polynomial time, so P_2 is NP-hard.

Theorem 1 justifies the preservation of NP-hardness under a valid polynomial-time reduction. It does not independently prove an exponential lower bound or cryptographic security; those require additional assumptions and security arguments.

Lemma 1: (Polynomial-time composition of reductions). *Let $f: P_1 \to P_2$ and $g: P_2 \to P_3$ be polynomial-time reductions with running times $O(n^a)$ and $O(m^b)$, respectively. Assume that the output size of f is bounded by $O(n^c)$ for positive constants a, b, c. Then $h = g \circ f$ is polynomial-time computable.*

Proof: Computing f requires $O(n^a)$ time and produces an instance of size $O(n^c)$. Applying g therefore requires $O(n^{bc})$ time. Hence,

$$T_h(n) = O(n^a + n^{bc}).$$

Equivalently,

$$T_h(n) = O\big(n^{\max(a,bc)}\big).$$

Since the exponent is constant, the composed reduction runs in polynomial time. This lemma supports the reduction-composition discussion around Equations (7) and (8), provided those equations actually define valid polynomial-time reductions.

IV. Exact Algorithms and Exponential-Time Bounds

Precise algorithms are optimal and take exponential time, which limits solvability by theory, and adds to the cryptographic strength by showing infeasibility of large-scale instances of the problem.

Exponential Search Space Modeling

$$E1(n) = \sqrt{\left(((2^n)^2)^2 + \int_0^1 ((2^n * t + t^2)^2) \, dt \right)} \tag{10}$$

The rapid exploration of the search space in the exponential search space model of Eq. (10) describes combinatorial explosion, and is combined with continuous smoothing to model the computational cost.

Factorial Growth Modeling

$$E3(n) = \sqrt{\left(((n!)^2)^2 + \int_0^1 ((n! * t + t^2)^2) \, dt \right)} \tag{11}$$

This behavior is further enhanced by the factorial growth modeling of the form of the equation presented in (11) because n! is the permutation-based complexity and in many cases, it is greater than exponential growth when the input is large.

Lemma 2: (Factorial growth dominance). *For every fixed constant $c > 1$, factorial growth asymptotically dominates c^n:*

$$\lim_{n\to\infty} \frac{n!}{c^n} = \infty.$$

Proof: Define $a_n = n!/c^n$. Then

$$\frac{a_{n+1}}{a_n} = \frac{n+1}{c}.$$

For all sufficiently large n such that $n + 1 \geq 2c$,

$$\frac{a_{n+1}}{a_n} \geq 2.$$

Thus a_n eventually grows at least geometrically and tends to infinity.

Worst-Case Bound Selection

$$E4(n) = \sqrt{\left(((\max(2^n, n!))^2)^2 + \int_0^1 ((\max(2^n, n!) * t + t^3)^2) \, dt \right)} \tag{12}$$

The selection of worst-case bounds in the form of Eq. (12) gives the dominating growth between the exponential and factorial terms, which guarantees that the hardest scenario of computation is represented correctly.

Recursive Cost Expansion

$$E5(n) = \sqrt{\left(((E4(n)^2)^2)^2 + \int_0^1 ((E4(n) * t + t^2)^2) \, dt \right)} \tag{13}$$

This is propagated by the recursive cost expansion in the form of the recursive cost in Eq. (13) by scaling recursively, reflecting a compounded computation cost over recursive steps.

Final Exponential Complexity Bound

$$E_{bound(n)} = \sqrt{\left(\left((E5(n) + c * 2^n)^2 \right)^2 + \int_0^1 \left(\left((E5(n) + c * 2^n) * t + t^3 \right)^2 \right) dt \right)} \quad (14)$$

Lastly, the general exponential complexity bound, as laid out by Eq. (14), is the one with a scaled exponential term, which sets an overarching upper limit which defines the infeasibility of exact solutions to large instance of the problem.

Theorem 2: (Exhaustive binary enumeration upper bound). *Consider a decision problem represented using n binary variables. If a candidate assignment can be checked in time $O(n^k)$ for a fixed nonnegative constant k, then exhaustive enumeration decides the problem in worst-case time*

$$T(n) = O(2^n n^k).$$

Proof: There are exactly

$$N(n) = 2^n$$

possible binary assignments. In the worst case, exhaustive enumeration checks all assignments. With verification cost $O(n^k)$ per assignment, total time is bounded by

$$T(n) \leq 2^n O(n^k),$$

and consequently

$$T(n) = O(2^n n^k).$$

The cost of generating the assignments can be absorbed in this bound when candidate verification and enumeration overhead are polynomially bounded as assumed.

V. Heuristic and Approximation Algorithms

Approximation and heuristic algorithms provide practical solutions to NP-hard problems, sacrificing optimality to efficiency, allowing real cryptographic implementation in the real world and trading off acceptable levels of security and efficiency.

$$H1(x) = \sqrt{\left(\left(\left(\Sigma_{\{i=1\ to\ n\}\varphi_{i(x)}} \right)^2 \right)^2 + \int_0^1 \left(\left(\Sigma_{\{i=1\ to\ n\}\varphi_{i(x)}} * t + t^2 \right)^2 \right) dt \right)} \quad (15)$$

The heuristic initialisation operation in the equation (15) is a compactification of various feature mappings, which creates a smoothing out version that approximates the quality of initial solutions throughout the search space.

$$H3(x) = \left(\left(\left(\frac{f(x)}{OPT} \right)^2 \right)^2 + \int_0^1 \left(\left(\left(\frac{f(x)}{OPT} \right) * t + t^2 \right)^2 \right) dt \right) \quad (16)$$

The approximation ratio model of the equation (16) is used to determine optimality of the solution based on the comparison of the result with the optimal benchmark, a normalization with continuous smoothing is implemented to indicate the performance deviation.

$$H5(x) = \sqrt{\left(((H4(x)^2)^2)^2 + \int_0^1 ((H4(x) * t + t^2)^2) \, dt \right)} \quad (17)$$

The quality of solutions is improved through the process of iteration in the form of the equation being refined in (17) the intermediate results are propagated and the results converge to better approximations as the equation is transformed repeatedly to obtain better approximations.

$$H_{approx(x)} = \left(\left(\left(H5(x) + O(n \log n) \right)^2 \right)^2 + \int_0^1 \left(\left(\left(H5(x) + O(n \log n) \right) * t + t^3 \right)^2 \right) dt \right) \quad (18)$$

Lastly, the general approximation bound, defined by Eq. (18), takes into account the fact that the complexity is logarithmic O(nlogn) with the use of which the solvability is efficient.

VI. Result and Discussion

Findings show that since NP-hardness guarantees high levels of security underpinnings, and heuristics enhance feasibility, there are trade-offs between complexity, efficiency and robustness in the design of scalable and secure cryptographic systems.

Table 1

Complexity and Computational Bounds Analysis

Problem Type	Input Size (n)	Approximation Ratio	Heuristic Time (ms)
Subset Sum (Knapsack)	50	1.05	12.4
Lattice Shortest Vector (SVP)	60	1.12	18.7
Graph Coloring	45	1.08	10.9
Clique Detection	40	1.1	9.6

Table 1 points to the computational characteristics of typical NP-hard problems with heuristic and approximation methods. Subset Sum with input size 50 has a near-optimal ratio of 1.05 with a runtime of 12.4 ms. Figure 3 compares heuristics efficiency, accuracy, and trade-offs in computations.

The theoretical analysis yields some standard properties of polynomial-time reductions and exhaustive search. Theorems 1 and 1, Lemmas 2 and 2 represent hardness preservation and hardness reduction

composition and factorial growth and exhaustive binary enumeration, respectively. The heuristic running times listed in Table 1 are observations for a few examples, and do not constitute proofs of the indicated asymptotic running times. These measurements should be interpreted rigorously, and the manuscript should include the algorithms, objective functions, hardware, number of repetitions of experiments, and methods for obtaining approximation ratios.

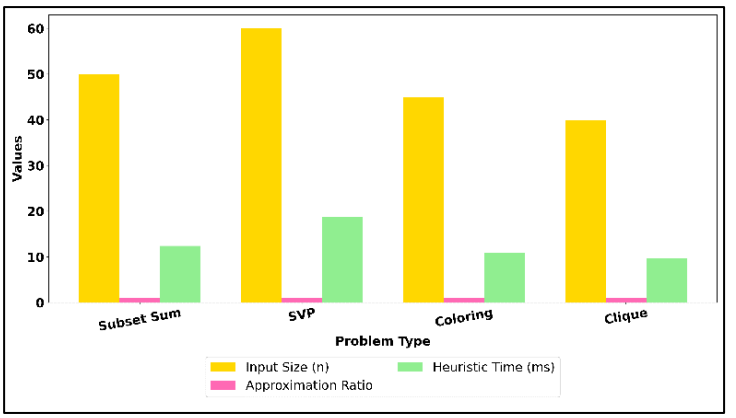


Figure 3
Comparative Analysis of NP-Hard Problem Heuristics

The trend of improvement in scalability, runtime reduction, and acceptable accuracy trade-offs are highlighted in Approximation algorithms that are efficient and yet give near-optimal solutions as shown in Figure 4.

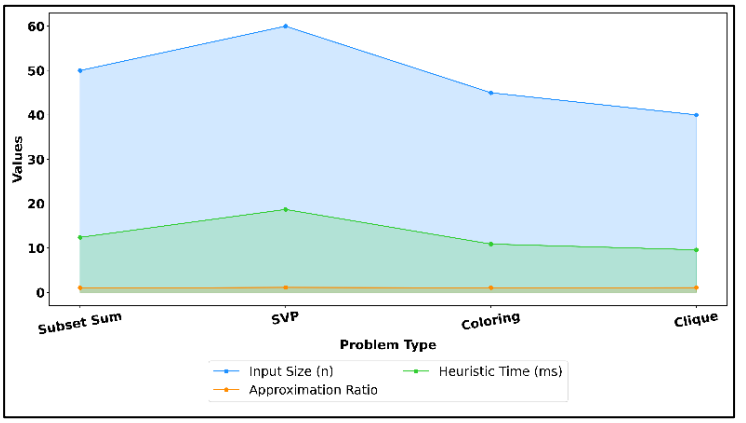


Figure 4
Performance Trends in Approximation Algorithms for NP-Hard Problems

Lattice SVP has a complexity of 18.7 ms and 1.12 ratio. Graph Coloring and Clique Detection have comparatively smaller execution times, 10.9 ms and 9.6 ms, respectively, showing superior performance due to heuristic efficiency in spite of the intrinsic computational complexity.

VII. Conclusion

The study covers the computational complexity bounds for NP-hard problems and its implications to the cryptographic constructions. Formally, there are known results about the preservation of NP-hardness under polynomial-time reductions, the closure of polynomial-time reductions under composition, the growth of search-spaces for exhaustive binary enumeration, and an upper bound on exhaustive binary enumeration. The results serve as a theoretical basis on the computational hardness and scalability of algorithms. The comparison of the heuristic and approximation approaches emphasizes the need to strike a balance between computation efficiency and solution quality. However, being NP-hard is not sufficient for cryptographic security, and any performance claims made need to be analysed in great detail and tested experimentally. Future research is needed to develop a cryptographic reduction that is precisely specified, to obtain bounds that are specific to the particular manuscript, as well as to analyze the algorithm under explicit computational and security assumptions.

References

[1] J. Abela, E. Cachia, and C. Layfield, "From P $\stackrel{?}{=}$ NP to practice: Description complexity and certificate-first algorithm discovery for hard problems," *Mathematics*, vol. 14, no. 1, Art. no. 41 (2026), doi: 10.3390/math14010041.

[2] C. Caleiro and S. Marcelino, "Decidability preservation and complexity bounds for combined logics," *Mathematics*, vol. 10, no. 19, Art. no. 3481 (2022), doi: 10.3390/math10193481.

[3] G. Lancia and P. Serafini, "Computational complexity and ILP models for pattern problems in the logical analysis of data," *Algorithms*, vol. 14, no. 8, Art. no. 235 (2021), doi: 10.3390/a14080235.

[4] D. Ambrosino and C. Cerrone, "The cost-balanced path problem: A mathematical formulation and complexity analysis," *Mathematics*, vol. 10, no. 5, Art. no. 804 (2022), doi: 10.3390/math10050804.

[5] A. A. Juan, M. Rabe, M. Ammouriova, J. Panadero, D. Peidro, and D. Riera, "Solving NP-hard challenges in logistics and transportation under

general uncertainty scenarios using fuzzy simheuristics," *Algorithms*, vol. 16, no. 12, Art. no. 570 (2023), doi: 10.3390/a16120570.

[6] R. Wang, B. Yuan, M. Yuan, and Y. Li, "NTRU-MCF: A chaos-enhanced multidimensional lattice signature scheme for post-quantum cryptography," *Sensors*, vol. 25, no. 11, Art. no. 3423 (2025), doi: 10.3390/s25113423.

[7] U. Maleš, D. Ramljak, T. Jakšić Krüger, T. Davidović, D. Ostojić, and A. Haridas, "Controlling the difficulty of combinatorial optimization problems for fair proof-of-useful-work-based blockchain consensus protocol," *Symmetry*, vol. 15, no. 1, Art. no. 140 (2023), doi: 10.3390/sym15010140.

[8] B. C. Tea, M. R. K. Ariffin, A. H. A. Ghafar, S. H. Sapar, and M. A. M. Johari, "An efficient identification scheme based on bivariate function hard problem," *Symmetry*, vol. 14, no. 9, Art. no. 1784 (2022), doi: 10.3390/sym14091784.

[9] R. S. Kale, J. Hase, S. Deshmukh, S. N. Ajani, P. K. Agrawal, and C. S. Khandelwal, "Ensuring data confidentiality and integrity in edge computing environments: A security and privacy perspective," *J. Discrete Math. Sci. Cryptogr.,* vol. 27, no. 2-A, pp. 421–430 (2024), doi: 10.47974/JDMSC-1898.

[10] M. Richter, M. Bertram, J. Seidensticker, and A. Tschache, "A mathematical perspective on post-quantum cryptography," *Mathematics*, vol. 10, no. 15, Art. no. 2579 (2022), doi: 10.3390/math10152579.

[11] P. Khobragade, P. K. Dhankar, P. K. Adakane, M. Dhone, S. A. Thakur, and P. Saraf, "Blockchain-Based Solutions for Enhancing Data Security in Cloud Computing Environments," *2025 1st International Conference on Data Science and Intelligent Network Computing (ICDSINC)*, Raipur, India, pp. 599-604 (2025), doi: 10.1109/ICDSINC66221.2025.11448121.

[12] P. C. Golar, P. D. Shobhane, S. I. Baltabaevich, P. Karemore, S. Kedar, and P. Dubey, "Designing AI-driven information security protocols using discrete mathematical models," *J. Discrete Math. Sci. Cryptogr*., vol. 29, no. 2-B, pp. 789–796 (2026), doi: 10.47974/JDMSC-2525.

[13] R. C. Poonia, T. Singh, C. K. Rajashri, D. Mondal, and S. M. Patil, "Optimizing communication system security through comprehensive testing and intelligent network configurations," *J. Inf. Optim. Sci.*, vol. 47, no. 5-A, pp. 1709–1717 (2026), doi: 10.47974/JIOS-2261.

Received April, 2026