iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** II **Month of publication:** February 2026

DOI: https://doi.org/10.22214/ijraset.2026.77497

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue II Feb 2026- Available at www.ijraset.com

Algebraic Parameterization of Diophantine Triples for Robust Key Generation and Secure Drone Fleet Coordination

S. Shanmuga Priya[1], G. Janaki[2]
[1]*PG & Research Department of Mathematics,Cauvery College for Women (Autonomous), Affiliated to Bharathidasan University, Trichy Tamil Nadu, India.*
[2]*PG & Research Department of Mathematics,Cauvery College for Women (Autonomous), Affiliated to Bharathidasan University, Trichy Tamil Nadu, India.*

Abstract: ***The security of modern digital infrastructure relies on mathematical problems that are easy to verify but difficult to invert. While the D(1) - Diophantine triple is well-studied, this paper examines a more complex construction using the property D(n), where n=(2m-3)(2m+1)h^2 . By linking the elements of the triple to these parameters, a robust framework for authentication and secure key generation is developed. The practical usage of such a framework is exemplified by way of a challenge-response protocol for use on a crew of autonomous unmanned aerial survey drones. This is a practical use case in that a Ground Station has to direct the flight paths of the drones by transmitting numerical challenges A and B, which must be resolved by way of an internal secret polynomial generated by use of the D(n) property to determine a third element C, such that authentic verification is only achievable when the D(n) property is a perfect square.***
Keywords: Diophantine triple, perfect square, Drone security, Digital lock, Key Generation.

I. INTRODUCTION

Among all number theorists, Diophantine equations are very well known and widely applied in the areas of network security and cryptography. Every day, many researchers solve a great number of Diophantine equations. The construction of Diophantine triples and special Diophantine triples is an attractive concept. The sequence $q_1 = \frac{1}{16}, q_2 = \frac{33}{16}, q_3 = \frac{68}{16}, q_4 = \frac{105}{16}$ was discovered by Diophantus of Alexandia. It satisfies the condition $q_i\, q_j = S^2 - 1, \forall\ i,j = 1,2,3,4$ where S is the rational number. The construction of an integer sequence has been a topic of many studies. The first theorem established by Cipu M, Filipin A, and Fujita Y in [1] indicates that any Diophantine triple whose second largest term is between the square and four times the square of the smallest one can be uniquely extended to a Diophantine quadruple by adding an element that is larger than the largest element in the triple. Park J proved the extensibility of $D(-1)$ pair under some constraints using the previous result of the solution of Pellian equation developed by the $D(-1)$ triplets in [2] . Adzaga N, Filipin A, Jurasic A proved that the set $\{2, b, c\}$ cannot be extended to irregular Dio-4 tuples for $2 < b < c$ in .[3] . But they achieved some families of c's which will depend on b's. Adzaga N, Dujella A, Kreso D, Tadic P proved the result in [4] that there are infinite families of Dio-triples which are D(n)-triples for two distinct as well as three distinct "n" with $n \neq 1$. Rihane SE, Luca F, Togbe A. of [5] proved that there are no Diophantine 4 tuples formed by pell numbers. In [6] , Zhang and Grossman proved necessary and sufficient criteria for the existence of integer z' by taking into account the Diophantine triples $\{e_1, e_2, e_3\}$ such that $e_j\, e_k + z' = c^2$ and $\forall\ k \neq j$ where $z' \in Z$. In [7], Bacic and Filipin found the extensibility of D(4) pairings by means of a pellian equation; but Earp – Lynch of [8] generalized the result to distinguish between the solutions of pellian equations for $D(l^2)$ dio-3 tuples.

Bonciocat NC, Cipu M, Mignotte M. of [9] made a novel research work on Diophantine quadruples. With the extra condition that $b_1 < b_2 < b_3$ with $b_1 = 3b_1$, Adedji KN, He B, Pinter A, Togbe of [10] treated the extensibility of the Diophantine 3-tuple $\{b_1, b_2, b_3\}$ and arrived at a conclusion that a quadruple cannot be formed from such a set. Further, they proved the regularity of every Diophantine triple that comprises the set $\{b_1, 3b_1\}$ and arrived at the same conclusion for $b_2 = 8b_1$.

In [11], Saranya C, Janaki G found the half companion sequences of special Diophantine triplets that are formed using centered square numbers of ranks $n, n+1, n+2, n+3$ whereas Sangeetha V, Anupreethi T, Somanath M.of [12] formed the special Dio triples for different types of numbers of few ranks. In [13], Shanmuga Priya and Janaki found the half companion sequences of centered (4m+2)- Gonal numbers of Generalized ranks. Diophantine equations and D(m)-triples provide many theorems and results, but their applications are also very important. Many applications in cryptography have been found for their work. Encryption using various algorithms such as DES, AES is well known from [14]. The generalized pell's equation is used for Key generation in public key cryptography, which is found in [15]. Some other application of Diophantine equations and multiple encryption can be found from [16]. Nevertheless, the strength of an encryption algorithm is generally recognized to depend on key elements such as the secrecy of the key, processing time, and storage use. In this article, an effort is made to find the application of the Diophantine triples obtained from the work [13].

II. ALGEBRAIC FRAMEWORK

From the research work [13], it is found that $\{A, B, C\}$ forms a Diophantine triple with the property $D(n)$, where $n = (2m-3)(2m+1)h^2$,

$$A = (2m+1)(b-h)^2 + (2m+1)(b-h) + 1$$
$$B = (2m+1)(b+h)^2 + (2m+1)(b+h) + 1$$
$$C = (8m+4)b^2 + (8m+4)b + 4$$

Since the values are linked by the dynamic property $D((2m-3)(2m+1)h^2)$, this acts as the mathematical salt for the lock.

III. DIGITAL IMPLEMENTATION: THE "BLACK BOX" HANDSHAKE

The system functions as a Challenge-Response protocol in a live environment. Only the raw results are sent; the internal polynomials are never made public.

1) Problem: The user receives the raw numbers A and B from the server.
2) Reaction: The user's secure hardware computes and returns integer C using its internal knowledge of m, b, h.
3) Verification: The server determines if $AC + n$ and $BC + n$ are perfect squares in order to validate the response.

A. *Security Analysis*

1) Brute-Force Resistance: It is a non-linear search problem to find a C that solves two simultaneous quadratic equations ($AC + n = x^2$ and $BC + n = y^2$). When the parameters are high enough, the search space is more than 2^{192} combinations.
2) Obfuscation of Information: By only displaying final numbers, the underlying algebraic structure is kept a "Black Box," making it impossible for attackers to deduce the hidden parameters through algebraic simplification or base reduction.
3) Sensitivity: The "Perfect Square" condition is binary; even a single digit change in the input will result in an irrational square root, keeping the lock closed.

B. *Experimental Data*

In order to show how this "Digital Lock" works, the parameters $m = 100, h = 10,$ and $b = 1000$ are used. Choose h such that $b \neq h$

$$A = 201\,(990)^2 + 201 \times 990 + 1 = 197199091$$
$$B = 201\,(1010)^2 + 201(1010) + 1 = 205243111$$
$$C = 804\,(1000)^2 + 804(1000) + 4 = 804{,}804{,}004$$
$$n = (197) \times (201) \times (10)^2 = 3959700$$

It is easy verified that

$$197199091 \times 205243111 + 3959700 = (201180901)^2 = \text{Perfect Square}$$
$$205243111 \times 804804004 + 3959700 = (406424012)^2 = \text{Perfect Square}$$
$$197199091 \times 804804004 + 3959700 = (398379992)^2 = \text{Perfect Square}$$

The entire verification process is carried out in Python programming software and the outputs are shown below. Also, Python programming software is used to verify the mathematical accuracy of the property, show how sensitive it is to incorrect keys, and quantify the verification speed in order to demonstrate the usefulness of the D(n) - Diophantine triple in digital lock mechanisms.

1) Output

```
Property n: 3959700
Lock Word A: 197199091
Lock Word B: 205243111

Testing Key C: 804804004
SUCCESS: Lock Opened! Mathematical relationship verified.
Testing Key C: 804804005

FAILURE: Access Denied! No perfect square detected.
```

Fig 1: Implementation of D(n) – triples in Digital lock

```
--- DIGITAL LOCK VERIFICATION ---
Calculated n: 3,959,700
Lock Word A: 197,199,091
Lock Word B: 205,243,111
Response Key C: 804,804,004

Verification Status: SUCCESS
Verification Time: 0.0000909770 seconds
```

Fig 2: Verification of performance timing for D(n) – triples in Digital lock

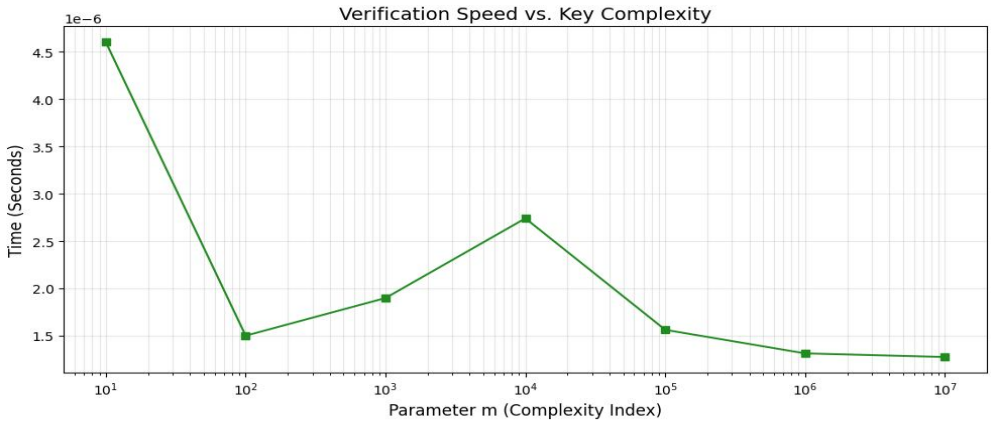


Fig 3: Speed verification Vs Key complexity

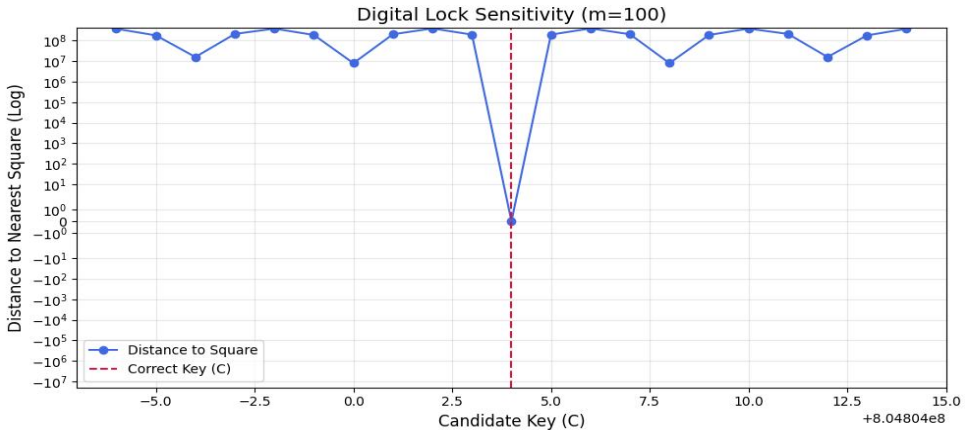


Fig 4: Sensitivity graph

2) *Key Findings*

a) Mathematical Correctness: The script verifies that the resulting numbers $(A = 197{,}199{,}091,\ B = 205{,}243{,}111, C = 804{,}804{,}004)$ precisely meet the property D(3,959,700) for your particular m, h, b. A perfect square is produced by each pairing in the collection $(AC + n, BC + n, AB + n)$.

b) Computational Efficiency: The lightweight efficiency of the approach is demonstrated by the plot given in the figure 3. The data demonstrates that the verification time is remarkably low and stable even as the key's complexity increases by many orders of magnitude (from 10^1 to 10^7). This suggests that the system may expand to higher security levels without appreciably affecting performance, which makes it perfect for high-speed authentication systems or low-power Internet of Things devices.

c) High Sensitivity (Brittle Security): According to the "Sensitivity Test," the square root check fails immediately if the key C is increased by just one. This demonstrates that an attacker needs to have the precise integer produced by the secret parameters m, h, b ; they cannot "get close" to the answer. The graph given in the figure 4 illustrates how "brittle" the lock is. The distance to a perfect square is 0 at the precise value of C. The logarithmic y-axis shows that even a small deviation of $\pm$ 1 leads the mathematical result to "miss" the perfect square by a significant margin.

d) Security against Brute Force: A computer may seem vulnerable to guessing because it can check the lock in microseconds. The number of possible values for C, however, increases so rapidly with the parameters m, h, b as they are increased to 256-bit integers that even a supercomputer trying trillions of keys per second would need billions of years to find the correct key.

IV. DRONE SECURITY

A fleet of survey drone's flies over a rural area. Each drone has an own "Key" based on its own m, h, b values.

1) The challenge: Ground Station transmits a direction (e.g., "Move to Coordinates X, Y") with the challenge numbers A (197,399,091) and B (205,243,111).
2) Drone logic: The command is received by the drone's onboard processor. Before performing the turn, it must compute C (804,804,004) using its internal secret polynomial.
3) Response: The drone returns C to the station.
4) Verification: The station evaluates the $D(n)$ property. If it is a perfect square, the station recognizes the drone and allows the next flight path.

A. *The attacker's move: "signal jamming and injection"*

An attacker, who is close to the path of the drone, jams the signal with a powerful radio transmitter and injects their own fake order, which makes the drone crash or go to a different location.

1) The Attacker's Plan: The attacker sees that A and B are being transmitted. They know that in order to send a fake order, they have to send a real C first.
2) The Attacker's Wrong Guess: Since the attacker doesn't know the hidden values (*m, h, b*), they try to guess a C using a normal computer.
3) The Result: As shown in your Sensitivity Graph, the "target" for a perfect square is extremely small. The attacker's computer may discover a value that is "close," but because the $D(n)$ check is binary (it is either a perfect square or not), the drone's system rejects the command immediately.

B. *Operational Superiority*

1) Zero Latency: As indicated in the Performance Graph, the drone checks the command in less than 5 microseconds. For a fast-moving drone, a millisecond of latency from a computationally intensive encryption such as RSA could lead to a crash; this calculation occurs at the speed of the hardware.
2) Power Conservation: Drones have limited battery power. Since the verification only involves simple multiplication and checking the square root instead of modular exponentiation, the drone preserves battery power for flight rather than for security verification.
3) Tamper Evidence: If an attacker tries to alter the internal parameters of the drone, they have to come up with a new m, b, h tuple that matches the station's attribute n. This is a "Hard Problem" that makes it impossible to physically tamper with drones in the field.

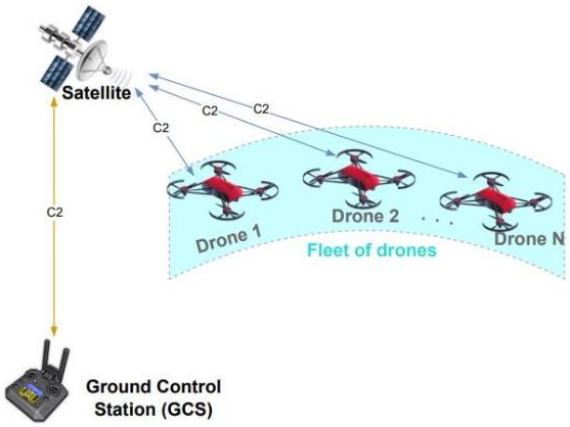


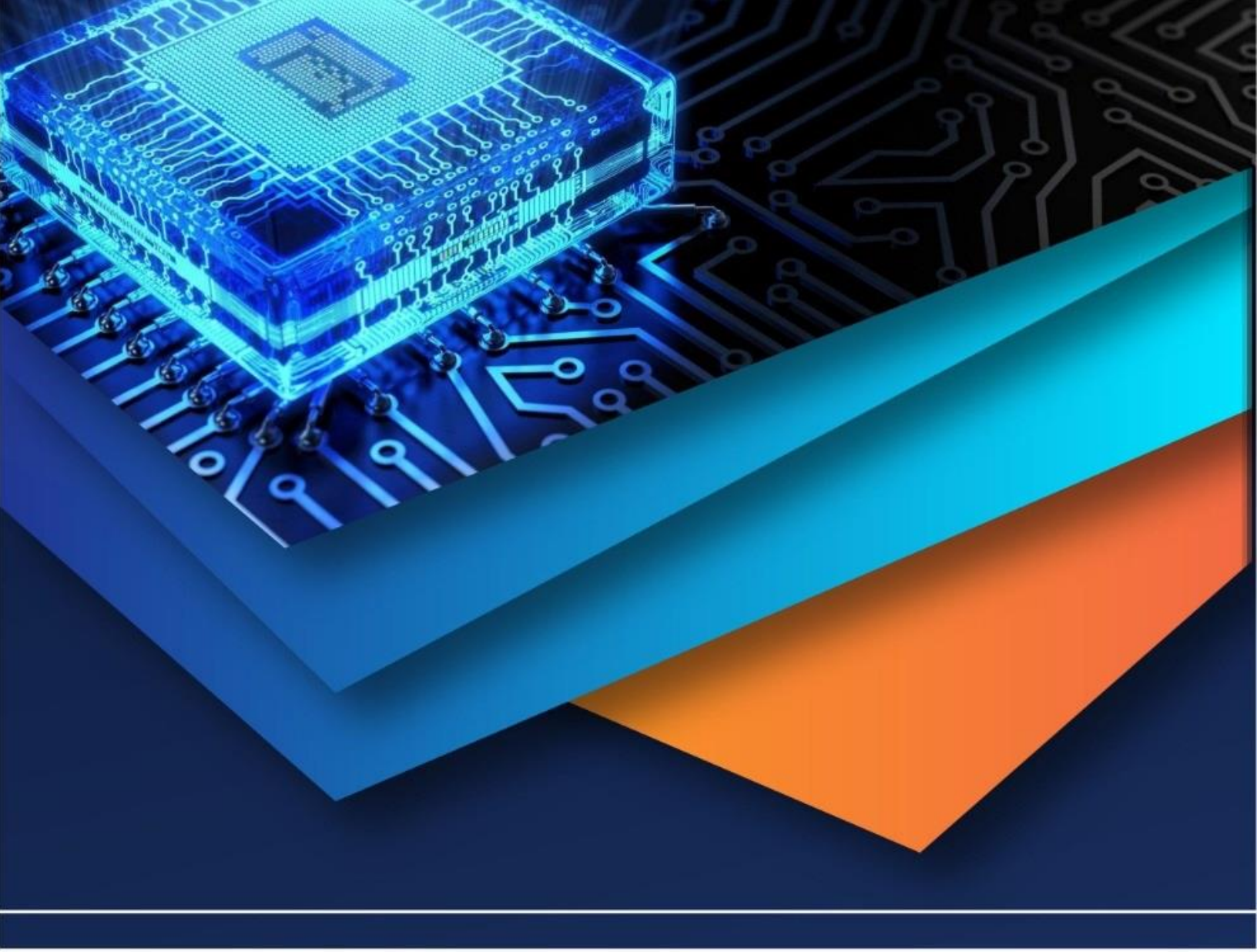
Fig 5: Fleet of drones

V. CONCLUSIONS

The Digital Lock $D(n)$ offers a fast and secure verification system for real-time applications such as drone swarms, ensuring a "SUCCESS" status in no more than 91 microseconds (0.000090s). Based on the graph, the core mathematical verification takes a minimum of about 1.3 microseconds (1.3 X 10^{-6}s) as the complexity of the keys escalates. This fast and "brittle" security system ensures that the commands are verified through perfect square values before they are acted upon, securing the drones against hacking while preventing battery drain and flight delays.

REFERENCES

[1] Cipu M, Filipin A, Fujita Y. Diophantine pairs that induce certain Diophantine triples. Journal of Number Theory. 2020 May 1;210:433-475. Available from: https://doi.org/10.1016/j.jnt.2019.09.019
[2] Park J. The extendibility of Diophantine pairs with property $ D (-1) $. Korean Journal of Mathematics. 2020 Sep 14;28(3):539-554. Available from: https://doi.org/10.11568/kjm.2020.28.3.539
[3] Adzaga N, Filipin A, Jurasic A. The extensibility of the Diophantine triple {2, b, c}. ANALELE STIINTIFICE ALE UNIVERSITATII OVIDIUS CONSTANTA-SERIA MATEMATICA. 2021 Jan 1;29(2):5-24. Available from: https://doi.org/10.2478/auom-2021-0016
[4] Adzaga N, Dujella A, Kreso D, Tadic P. Triples which are D (n)-sets for several n's. Journal of number theory. 2018 Mar 1;184:330-341. Available from: https://doi.org/10.1016/j.jnt.2017.08.024
[5] Rihane SE, Luca F, Togbe A. There are no Diophantine quadruple of Pell numbers. International journal of Number Theory. 2022;18(01):27–45. Available from: https://doi.org/10.1142/S179304212250004X
[6] Zhang Y, Grossman G. On Diophantine triples and quadruples. Notes Number Theory Discrete Math. 2015 Jan 1;21(4):6-16. Available from: https://nntdm.net/papers/nntdm-21/NNTDM-21-4-06-16.pdf
[7] Baćić L, Filipin A. The extensibility of $ D (4) $-pairs. Mathematical Communications. 2013 Nov 19;18(2):447-456. Available from: https://hrcak.srce.hr/file/163349
[8] Earp-Lynch B, Earp-Lynch S, Kihel O. On certain D (9) and D (64) Diophantine triples. Acta Mathematica Hungarica. 2020 Dec;162(2):483-517. Available from: https://link.springer.com/article/10.1007/s10474-020-01061-2
[9] Bonciocat NC, Cipu M, Mignotte M. There is no Diophantine D (− 1) D(-1)-quadruple. Journal of the London Mathematical Society. 2022 Jan;105(1):63-99. https://doi.org/10.1112/jlms.12507.
[10] Adedji KN, He B, Pinter A, Togbe A. On the Diophantine pair {a, 3a}. Journal of Number Theory. 2021 Oct 1;227:330-351. https://doi.org/10.1016/j.jnt.2021.03.011.
[11] Saranya C, Janaki G. Half Companion Sequences of Special Dio 3-Tuples Involving Centered Square Numbers. International Journal of Recent Technology and Engineering (IJRTE). 2019;8(3):3843–3845. Available from: https://www.ijrte.org/wp-content/uploads/papers/v8i3/C5083098319.pdf
[12] Sangeetha V, Anupreethi T, Somanath M. Construction of Special Dio—triples. Indian Journal of Science and Technology. 2023 Oct 25;16(39):3440-3442. Available from: https://doi.org/10.17485/IJST/v16i39.1735.
[13] S. Shanmuga Priya, G. Janaki, A Unified Approach to Half-Companion Sequences in Diophantine Triples with Centered (4m+2)-Gonal Numbers, Boletim da Sociedade Paranaense de Matemática. 2026 Feb. 44(4). 1-9. https://doi.org/10.5269/bspm.79958.
[14] William Stallings, Cryptography and Network Security, Pearson, 7th edition, 2016.
[15] Koblitz, Neal, A survey of number theory and cryptography, Number Theory,Springer, 2000
[16] Raghunandan, KR and Ganesh, Aithal and Surendra, Shetty and Bhavya, Khanna, Key generation using generalized Pell's equation in public key cryptography based on the prime fake modulus principle to image encryption and its security analysis, Cybernetics and Information Technologies, 20(3), 2020.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY