Egyptian Informatics Journal 24 (2023) 71–80

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Community oriented socio-behavioural PentaPlicative Cipher Technique

Siya Garg [a], Vinita Jindal [a], Harshit Bhatia [b], Rahul Johari [c,*], Shrey Gupta [c]

[a] Department of Computer Science, Keshav Mahavidyalaya, University of Delhi, Delhi 110034, India
[b] Reval India Private Limited, Gurugram 122007, Haryana, India
[c] SWINGER: Security, Wireless, IoT Network Group of Engineering and Research Lab, University School of Information, Communication and Technology (USICT), Guru Gobind Singh Indraprastha University, Sector-16C, Dwarka, Delhi 110078, India

ARTICLE INFO

Article history:
Received 5 May 2022
Revised 18 October 2022
Accepted 1 December 2022
Available online 4 January 2023

Keywords:
Community
Choice of interest
Cipher text
Cryptography
Decryption
Encryption
Message
Node
Point of Interest
Plain text
PentaPlicative
Social group
Symmetric key

ABSTRACT

Security of the data is of utmost importance, whenever the data flows on the network. In the current times and t[i]mes to come, cyber security is going to occupy the center stage in the lifecycle of any software. Security is an evergreen and everlasting area, because of the continuous threat from Hackers and Crackers. The proposed work focuses on the protection of the data in the area of Social Network. It has been achieved by the design and development of a new encryption technique called as PentaPlicative Cipher Technique. The PentaPlicative Cipher Technique makes use of multiple keys such as Latitude, Longitude, IP Address and MAC Address of node et al. to securely encrypt the message that needs to be transmitted through a non-secure channel. The paper concludes with an effective comparison in terms of space and time Complexity between Triplicative and PentaPlicative Cipher Technique.

1. Introduction

A community is defined as a group or cluster of social groups in a given network. A community usually builds up around interests and concerns the people share commonly amongst them. Such common point of interest or choice of interest are usually referred to as a 'base station' in context of community buildup and formation. The place of location of a single centralized base station represents a point of interest (POI) or Choice of Interest (COI). The number of communities or Socially Inclined Groups that build-up in a network is always equal to the number of base stations in the Social Network. In a social network the number of Social nodes would always be greater than the number of base stations. At initial stage, all base station are member node of its predefined community, here in this case a Social Group. A human carrier is likely to be part of COI/ POIs. It is calculated based on the frequency of the visits made near the base station. An exchange of message is performed as and when a carrier node comes in direct contact with its base station located near its POIs. This is predefined based on its social interest of the node. In the current research work, a new and innovative 'PentaPlicative Cipher technique(enhanced version of Triplicative Cipher Technique) with a key sharing algorithm has been introduced for the first time. This new superior technique has been designed to achieve automated and seamlessly integrated cryptosystem that ensures end-to-end message delivery system(to-fro) between node and base station in a SRS(Safe, Robust & Reliable, Secure) fashion. The PentaPlicative Cipher Technique employs a set of mathematical operations that adopts selected keys (denoted as K1, K2, K3, K4, K5) for encryption. As a sample case study, an example has been showcased in the current research work, where in the sender encrypts a string called '**SOCIAL**'. For depicting the process of encryption and decryption, for ease of

* Corresponding author.
E-mail addresses: vjindal@keshav.du.ac.in (V. Jindal), rahul@ipu.ac.in (R. Johari).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.12.001

showing mathematical operations, and for quick and effective simulation, smaller keys have been used in the case study with values ranging from 17 to 147(used in random order). However, in real life, these keys are much larger.

1.1. Abbreviations and acronyms

Abbreviations and acronyms that are used are defined in Table 1.

2. Problem statement

Security of the data is a critical aspect especially when an e-commerce or m-commerce e-retailer/aggregator intends to carry out the secure commercial financial transactions on the web. To achieve this, a secure, strong, robust, dynamic and adaptable cryptographic technique is the need of an hour. Well known traditional symmetric and asymmetric cryptographic techniques such as Caesar Cipher, Vigenere Cipher, Rail Fence, Affine Cipher, Hill Climbing Cipher, Additive Cipher et al. evolved over the years with aim to provide security of the data, but all of them possessed limitations in terms of the number of key used by them in the mathematical equations. This problem has been addressed in the current research work by proposing a PentaPlicative Cipher Technique that employs a set of mathematical operations that adopts set of five uniquely selected keys (denoted as K1, K2, K3, K4, K5) for cryptography operations.

3. Securing a community: cryptographic approach

The communication channel between the nodes and the base station is not secure and a sniffer could easily intercept the messages being exchanged between the member nodes in a community. The messages are being exchanged in a plain-text format without any kind of encryption on those messages. Hence, if intercepted, these messages would be visible to the sniffer. To make the channel secure, a cryptographic technique is required which would be employed to encrypt the messages before transmitting them on the unsecured channel. The proposed cryptographic approach exploits the preexisting community structure of a social network. The context-awareness of the node is utilized to draw location aware based efficient delivery of the messages. The attributes defined for the context awareness is denoted by set of characteristics of the system proposed namely metadata of the node's community, remaining buffer space and maximum buffer space.

A node may exploit the context information for efficient message delivery. When a node has a message that has to be delivered to the destination, it scans all the nodes in its neighborhood. If a destination node is found, a message is delivered to the destination node, else a next-hop node is selected. A node forwards a message only to that neighboring node whose delivery probability of message is higher than the node itself and is also more amongst all its neighboring nodes. Regularly, each node determines its probability to deliver a message concerning each destination. The proposed approach aims to secure message transmission between a node to node as well as node to the base station.

The communication is happening in real-time, hence the cryptographic technique needs to be fast without compromising on security. For this purpose, symmetric-key cryptography [1] is chosen which would be prompt and would also be secure with its private keys. The process of encrypting the message at the sender side, as well as decrypting the message on the receiver side would employ the use of the same private keys which would be available to both sender and receiver. Since the keys are private so only the sender and receiver would know about them and any third party (such as the sniffer) would not have access to them. In a secure cryptosystem [2], it should not be possible to decrypt the message without the correct private keys.

Table 1
List of Notations.

S.No.	Abbreviations	Full Form
1	PT	Plain Text
2	CT	Cipher Text
3	COI	Point of Interest
4	POI	Choice of Interest
5	SHA	Secure Hash Algorithm
6	CnC	Continent Code
7	CC	Country Code
8	SC	State Code
9	DC	District Code
10	TC	Tehsil Code
11	VC	Village Code
12	FC	Family Code
13	PC	Person Code
14	ECC	Eliptic Curve Cryptography

4. Literature survey

The authors of [3] conducted a thorough review of the existing cloud-based security and privacy models for storing, processing, and accessing Electronic Health Records(EHR).The investigated literature employs a variety of Attribute-Based Encryption (ABE) variants. It is indeed an excellent technique; however, excessive computations due to bi-linear operations affect its performance. The authors claim that looking for a solution to this problem would be a good research topic. In addition, the authors believe auditing may be quite beneficial. Further,the privacy of not only patients but of all stakeholders should be considered. They also proposed an architectural framework that is controlled only by the patient, wherein a patient obtains full authentication via hospital. He chooses his Medical Officer (MO), who accesses the records by using the Access Control List (ACL) security model, and the type of activity to be conducted is controlled by the Mandatory Access Control(MAC) model. In the event of an emergency, the architecture enables proxies to provide room for prompt assistance to any patient.

One of the greatest challenges in data exchange is ensuring the security of data transfer. The authors of [4] proposed a novel approach based on the combination of cryptography and steganography and aimed to make the cryptographic aspect of it even more impenetrable than before. The ciphertext generated using AES-HMAC is rearranged using a shifting algorithm and the resultant is a shifted ciphertext, which is then embedded into a stego cover for the transmission.

The process of enhancing mobile network security measures is ongoing. To provide a strong trust-based authentication mechanism for adapting to the often changing topology and validation of the new members in serverless computing, the authors of [5] suggested a new authentication-based scheme specifically intended for the MANET nodes using session token with fingerprint, and MAC address validation.

For the Lorenz systems, a generic scaling, reflection, rotation, translation, or shearing transformation of chaotic systems was suggested and demonstrated by the authors in [6] Applying transformations to the generated time series in this approach eliminates the need for post-processing. The chaotic system's differential equations contain six variables that affect the behavior of the system and increase its sensitivity. A presentation on trajectory control of the attractor's dynamic motion that explored several trajectories was also made. Further, key space is increased by the transformed Lorenz system in an image encryption system that passes the benchmark performance tests.

The biggest barrier to the widespread adoption of Internet of Things (IoT) technologies is still the security of smart devices.To mitigate attacks on sensor nodes, the authors of [7] have suggested a hash key-based management system for cluster networks. The establishment of a secure key between one-hop and multi-hop nodes is accomplished using a one-way hash function, and communication between the nodes is possible through a secure channel. The coordinator node distributes the local key to each node in the network, and the pair-wise key is generated inside the cluster for single hop nodes and outside the cluster for multi-hop nodes. Each node in the network has both of these keys. The proposed method was employed to prevent the jamming attack and was shown to be effective in reducing the attack's impact. Unauthorized access to sensitive data is one of the biggest issues facing the world today. This necessity becomes more vital in the healthcare sector because of the vulnerability of patients' data security and privacy. The authors of [8] developed an iris-based cancelable biometric cryptosystem that is impenetrable and eliminates the need to save the cryptographic key. Instead, the cryptographic key is retrieved during run time following the successful authentication of the user using the iris template. The system operates in two phases: encryption, which uses an iris image, an encryption key, and health records to produce helper data, a cryptographic hash, and encrypted health records that are saved on a smart card. The iris image template is created once more for decryption, and it is combined with helper data to generate the decryption key. A match is made between the cryptographic hashes of the encryption and decryption keys. If match is successful, generated decryption key is used to decrypt the encrypted health records.

The authors of [9] presented a comprehensive review of the security and privacy concerns of electronic health record systems. Electronic Health Records (EHR) do indeed make the sharing and management of records more effective. However, there are numerous concerns with EHR security, which prevent the use of these systems. The authors strongly advise that the latest EHR records should be encrypted with an efficient encryption technique that is simple to use by both patients and healthcare professionals to lower the barrier associated with security and privacy concerns and hasten the adoption of EHRs by users and providers of healthcare. Users all around the world use a multitude of devices to communicate, and the effectiveness of that communication is largely dependent on the network, which is prone to frequent network outages in harsh conditions. Authors in [10] have devised a strategy to overcome these obstacles. A community clustering routing algorithm was proposed wherein different network nodes were divided into communities using the k-modes algorithm; communities were then further clustered and merged based on information entropy, and a dynamic updating strategy was presented to assure information transmission efficiency.

It is critical to foresee friendships between small circles produced for new products and huge circles formed for mature products in order to draw clients from large mature product communities to the new one. The Collaborative Combined Link Prediction Algorithm (CCLPA) is proposed in [11], which extensively extracts user attention concentration (AC) features and overcomes the scale-free network's fluctuation of algorithm accuracy. A literature review of a Local Energy Community (LEC) integrated in a distribution network was presented by the authors in [12]. The review discussed all three layers of an LEC integration: grid, controller, and market, in detail and presented three different approaches for LEC integration where in these layers interact with one another in a simulation environment.

In [13] contact tracing has been leveraged to contain the spread of COVID-19. The authors proposed TraceMe, a solution that employed Mobile and Wireless Networks (MWNs) for contact detection and tracing with Online Social Networks (OSNs) for identifying potential patients and taking appropriate measures to avoid the spread of disease. Network science approaches were used in [14] to identify influential people and communities involved in public discussions such as allowing women to drive in Saudi Arabia. These methods, unlike machine learning methods, assess these discussions qualitatively.

Authors in [15] used intelligent optimization methods and artificial intelligence algorithms to build a reliable mobile wireless sensor network suitable for the complicated environment of social networks. They addressed reliability issues such as mobile path optimization on data collection efficiency and network reliability, reliable data transmission based on data fusion methods, and intelligent fault tolerance of multipath routings.

The most serious challenge to the viability of an Online Social Network (OSN) is data security and privacy. In [16], the authors reviewed existing OSN security and privacy solutions and made a compelling case for employing deep learning as intelligence security and blockchain for decentralizing privacy in both client–server and peer-to-peer models.

As even more individuals use social media to interact, it's becoming easier to de-anonymize users utilising side-channel data. In [17], the authors developed an attack architecture based on two attack vectors: device system states to a social network (DS-SN) and cross-social network correlation (SN-SN). For the DS-SN attack, a malicious app retrieves device information such as memory usage, CPU usage, and network data from victim's device followed by profile matching using learning models to match multiple social network profiles connected to the same account for the SN-SN attack.

The organization's InfoSec (Information Security) leadership structure was investigated and appraised by the authors in [18]. They concluded that any employee, regardless of their position in the organization, could become an influential InfoSec opinion leader and influence their colleagues' InfoSec behavior if they are regarded to have social power.

The authors of [19] provided a brief introduction to various cryptographic algorithms, ranging from historical ones like the Caesar cipher to Fully Homomorphic Encryption (FHE), to lessen the impact of assaults like Brute-force and cipher-text attacks, among others.

Instead of using mathematical and optimization techniques, which have several drawbacks such as computational load and inherent weaknesses, the authors of [20] used random selection-based substitution box structures as a defense against application attacks. The approach aimed to improve the nonlinearity criterion while maintaining speed and convenience of use.

In[21], the authors presented SCENERY, a novel, lightweight block cipher based on feistel structure. SCENERY employs bit-slice techniques, enabling the use of inexpensive hardware and efficient software implementation. The algorithm uses a binary matrix and is able to reach full dependency after four cycles. The performance of SCENERY is optimized for both hardware and software when compared to other encryption techniques.

CRYPOMPK was suggested by the authors of [22] to offer granular server application protection. The algorithm works in three parts; in the first part, the algorithm tracks and labels sensitive memory buffers and operations using information flow analysis. It then partitions the source code into crypto and non-crypto domains, and later it uses Memory Protection Keys (MPK) to safeguard secret keys from memory disclosure attacks.

The authors of the article [23] proposed a hybrid cryptography technique that combined the best aspects of symmetric and asymmetric algorithms. The proposed approach uses AES for encrypting the message(plain text) followed by key encryption using ECC and generates a 256-bit message digest using SHA256. The hybrid algorithm considerably speeds up the encryption and decryption of text files compared to existing algorithms; however, it takes slightly longer for image files.

5. Community based socio-behavioral cipher technique

A secure cryptosystem should serve the four underlying objectives [1]:

1. Confidentiality: The 'confidentiality' means that the transmitted information can only be understood by the intended recipient and any other sniffer may not be able to understand the original message.
2. Authentication: The 'authentication' of identities of both the sender and the receiver can be confirmed by either of the two.
3. Integrity: The 'integrity' of the message is said to have been withheld, if the transmitted message cannot be altered while it is being transmitted or even during the stale state of storage without detecting the alteration.
4. Non-repudiation: The 'non-repudiation' property points out the assurance that the sender cannot deny the validity of the sent message and the sender also cannot deny the authenticity of their key at any stage in the cryptosystem.

To have a secure cryptosystem, the cryptographic technique that will be used is the PentaPlicative Cipher Technique which fulfills the above enlisted basic objectives of a cryptosystem. The PentaPlicative Cipher Technique is a symmetric cryptographic technique, which means that the process of encrypting the plaintext message as well as decrypting the received cipher text uses the same set of private keys which are only known to the sender and receiver. The PentaPlicative Cipher Technique makes use of five private keys to securely encrypt the message that needs to be transmitted over an unsecure channel. Most cryptosystems usually only consider encrypting the message to be sent over the unsecure channel with pre-known private keys. However, they don't define the protocol or the mechanism via which the keys would be shared between the sender and receiver in a secure medium so that no sniffer would grab hold of them. The cryptosystem being introduced in the text that follows also gives a detailed explanation of the key sharing mechanism between the sender and the receiver. An introduction to the key-sharing mechanism here is necessary since the entire communication between the node and the base station is over an insecure channel. Hence, we need a mechanism through which the keys can also be shared discreetly between only the sender and the receiver, without the sniffer getting any knowledge of the keys.

6. Pentaplicative cipher technique

The PentaPlicative Cipher Technique [24] was first introduced as an enhanced successor of the Triplicative Cipher Technique [25]. The PentaPlicative Cipher Technique accepts a set of five private keys and using them performs a series of mathematical operations on plain text and returns a cipher text as output which is then transmitted further to the receiver. The receiver then uses the reverse mathematical operations on the received text by making use of the same set of five private keys to obtain the intended plain text message.

The original PentaPlicative Cipher Technique used a set of five keys that were mapped to the user's social information. However, to limit the user's interaction with the cryptosystem the keys are pre-selected from the system itself. When the node wants to interact with the base station and vice versa, the communication process should be seamless and must be robust enough to function independently without waiting for an input for the keys. Only the plain text to be transmitted should be provided as an input and the system should be intelligent enough to calculate the set of keys to be used automatically and must also share the same set of private keys with the receiver securely so that the process of decryption can be completed at the receiver end seamlessly. Keeping the same principle in light, the PentaPlicative Cipher technique was enhanced, and this new modification was adapted for making the end-to-end message delivery system between node and base station and vice versa an automated and seamlessly integrated cryptosystem.

6.1. Encryption operation

The PentaPlicative Cipher Technique employs a set of mathematical operations which are performed on the above-selected keys (denoted as K1, K2, K3, K4, K5) for encryption. The encryption steps which are carried over on the plain text (P) are briefed as mathematical equations below which result in the final cipher text (C):

- C1 = (P XOR K1)
- C2 = (C1 + K2) mod 256
- C3 = (C2 * K3) mod 256
- C4 = (C3 - K4) mod 256
- C5 = (C4 XOR K5)
- C = Bit_Dispersion (C5)

The intermediate cipher texts obtained at every step are denoted as C1, C2, C3, C4, and C5. These intermediate steps give a mathematical output with every corresponding key. Bit_Dispersion function is used to change the size of the input character set (plain text). It expands the size of the plain text of length 'n' to a cipher text with size 'm' where n < m. It does so by converting each character's ASCII into binary equivalent and then grouping the 8 bits of binary bits together. Then it makes a group of 6-bits each and converts it into the corresponding ASCII value hence dispersing the character count. This makes it impossible to map the characters of plain text to cipher text. The final text received as the output of this operation is transmitted as the Cipher Text from sender to receiver. The encryption process is briefed below as the algorithm:

- Step 1: start
- Step 2: accept plain text from user → PT
- Step 3: load keys K1, K2, K3, K4, K5
- Step 4: convert PT to ASCII decimal → P
- Step 5: XOR(P, K1) → C1
- Step 6: Add(C1, K2) → C2
- Step 7: Multiply(C2, K3) → C3
- Step 8: Subtract(C3, K4) → C4
- Step 9: XOR(C4, K5) → C5
- Step 10: Bit_Dispersion(C5) → C
- Step 11: return final Cipher Text as ASCII character → CT
- Step 12: end

6.2. Decryption operation

The receiver receives the cipher text (C) and then the receiver uses the same set of mathematical operations in reverse order with the same set of private keys to obtain the original plain-text message (P). The mathematical operations being used in the decryption can be easily depicted by a series of mathematical equations as shown below:

- D1 = Reverse_Bit_Dispersion (C)
- D2 = (D1 XOR K5)
- D3 = (D2 + K4) mod 256
- D4 = (D3 * K3-1) mod 256; K3-1 is the multiplicative inverse of the Key K3.
- D5 = (D4-K2) mod 256
- P = (D5 XOR K1)

The Reverse_Bit_Dispersion method changes the size of cipher text to match that of the plain text size by reducing the size of the cipher text of size ‘m’ to equate with the plain text of size ‘n’ where n < m. This is done by converting each character’s ASCII into binary equivalent and then grouping the 6 bits of binary bits together. Then it makes a group of 8-bits each and converts it into the corresponding ASCII value hence gaining back the original dispersed character count. The detailed algorithm for the decryption process has been briefed below:

- Step 1: start
- Step 2: accept cipher text received from sender →CT
- Step 3: load keys K1, K2, K3, K4, K5
- Step 4: convert CT to ASCII decimal → C
- Step 5: Reverse_Bit_Dispersion (C) → D1
- Step 6: XOR(D1, K5) → D2
- Step 7: Add(D2, K4) → D3
- Step 8: Multiply(D3, K3) → D4
- Step 9: Subtract(D4, K2) → D5
- Step 10: XOR(D5, K1) → P
- Step 11: return original Plain Text as ASCII character → PT
- Step 12: end

The decryption process once completed on the receiver’s end yields the plain-text message that the sender initially wanted to send. The real-time communication that is happening between the nodes and the base station in a community can be made secure by using this above entailed cryptographic technique.

7. An example of securing a plaintext message

Plaintext message that the sender needs to send is “SOCIAL”. For depicting the process of encryption and decryption and for ease of showing mathematical operations, smaller keys have been assumed. However, in real life, these keys are much larger. The keys assumed here are as follows:

K1 = 23, K2 = 31, K3 = 19, K4 = 17 and K5 = 147

Tables 2–5 show the detailed encryption as well the decryption operations which happen on both the sender as well as the receiver’s end.

8. The enhanced Diffie-Hellman key sharing algorithm

The PentaPlicative Cipher Technique is a keyed symmetric key cipher. This means that it requires the five private keys to be first securely shared between the sender and the receiver in a secure manner such that no third person can get access to these private keys. A mechanism needs to be in place to overcome the problem of the transmission of keys over an unsecured channel between sender and receiver. To overcome this problem, a public key cryptosystem is employed to share the keys between sender and receiver before the encryption process of PentaPlicative Cipher Technique can be initiated on the sender’s end. The key sharing process in our case employs an enhanced Diffie-Hellman Key sharing algorithm with added security. The enhanced Diffie-Hellman Key Sharing algorithm has been briefed in the text that follows.

The Key Sharing Algorithm makes use of two large prime numbers chosen such that the second number is the primitive root modulo of the first number. These two large numbers can be transmitted over an unsecure channel since they can be public and any sniffer having the knowledge of these two numbers will still not be able to figure out the actual key. These two public keys can be denoted as ‘p’ and ‘g’. The sender then chooses a large number ‘s’ which is kept secret and is never shared by the sender, and similarly, the receiver also chooses a large number ‘r’ which is kept secret by the receiver. Both the numbers ‘s’ and ‘r’ are never transmitted on the network and will always be private to the sender and receiver but never shared amongst each other.

There is a third private key ‘x’ which is a shared secret key between the sender and the receiver. This shared key is a session key which is generated by the server and distributed amongst the clients and this session key is unique for every session. The unique session key helps to prove the identity of both the sender as well as the receiver. The proof of identity is important because without the verification of both the sender and receiver, any third party or sniffer may alter the communication channel by sending data from his end and there would be no way to authenticate whether the data is coming from the authorized entities (valid sender/receiver) or not. Hence, this shared session key plays an important role in validating and verifying the identities of sender and receiver, and thus enhances the security of the traditional Diffie Hellman algorithm.

9. MAthematics behind enhanced-diffie hellman

The sender calculates a value ‘S’ by using the mentioned equation below, which it sends over to the receiver.

$$S = ((p-1).g)^{s.x} modp \tag{1}$$

The receiver calculates a value ‘R’ which it sends over to the receiver by using the mentioned mathematical equation:

$$R = ((p-1).g)^{r.x} modp \tag{2}$$

The values ‘S’ and ‘R’ are not the actual keys but are the values that would be used by both the sender and receiver to calculate the actual key ‘K’ which would be same for both sender and receiver and is the actual key that had to be transmitted. The sender calculates the key K as,

Table 2
Encryption Table.

P	C1= (P XOR K1)	C2=(C1 + K2) mod 256	C3=(C2 * K3) mod 256	C4=(C3- K4) mod 256	C5 = (C4 XOR K5)
S(83)	(83 XOR 23)=68(D)	(68 + 31) mod 256 = 99 (c)	(99 * 19) mod 256 = 89 (Y)	(89–17) mod 26 = 72 (H)	(72 XOR 147) = 219 (■) *Block, Graphic character*
O(79)	(79 XOR 23) = 88 (X)	(88 + 31) mod 256 = 119 (w)	(119 * 19) mod 256 = 213 (F)	(213–17) mod 256 = 196 (–)	(196 XOR 147) = 87 (W)
C(67)	(67 XOR 23) = 84 (T)	(84 + 31) mod 256 = 115 (s)	(115 * 19) mod 256 = 137 (ë)	(137–17) mod 256 = 120 (x)	(120 XOR 147) = 235 (δ)
I(73)	(73 XOR 23) = 94 ($\wedge$)	(94 + 31) mod 256 = 125 (})	(125 * 19) mod 256 = 71 (G)	(71–17) mod 256 = 54 (6)	(54 XOR 147) = 165 (N)
A(65)	(65 XOR 23) = 86 (V)	(86 + 31) mod 256 = 117 (u)	(117 * 19) mod 256 = 175 (>>)	(175–17) mod 256 = 158 (R)	(158 XOR 147) = 13 (♪)
L(76)	(76 XOR 23) = 91 ([)	(91 + 31) mod 256 = 122 (z)	(122 * 19) mod 256 = 14 (ψ)	(14–17) mod 256 = 253 (^{2})	(253 XOR 147) = 110 (n)

Table 3
Bit Dispersion Operation.

Obtained C_5	219	87	235	165	13	110		
C_5 binary	11011011	01010111	11101011	10100101	00001101	01101110		
Cipher	110110	110101	011111	101011	101001	010000	110101	101110
Cipher Text	54	53	31	43	41	16	53	46

Final transmitted Cipher text for CIPHER plaintext is 65▼+) ► 5.

Table 4
Reverse Bit Dispersion Operation.

Obtained C	54	53	31	43	41	16	53	46
C in binary	110110	110101	011111	101011	101001	010000	110101	101110
Re-dispersed	11011011	01010111	11101011	10100101	00001101	01101110		
Dispersed Text	219	87	235	165	13	110		

Dispersed text to be used to obtain plaintext is ■WδN♪n *Please note that ■ represents the ASCII block character.*

Table 5
Decryption Table.

D1	D2= (D1 XOR K5)	D3= (D2 + K4) mod 256	D4= (D3 * K3-1) mod 256	D5= (D4- K2) mod 256	D6 = (D5 XOR K1)
219 (■)	(219 XOR 147) = 72 (H)	(72 + 17) mod 256 = 89 (Y)	(89 * 27) mod 256 = 99 (c)	(99–31) mod 256 = 68 (D)	(68 XOR 23) = 83 (S)
87 (W))	(87 XOR 147) = 196 (–)	(196 + 17) mod 256 = 213 (F)	(213 * 27) mod 256 = 119 (w)	(119–31) mod 256 = 88 (X)	(88 XOR 23) = 79 (O)
235 (δ)	(235 XOR 147) = 120 (x)	(120 + 17) mod 256 = 137 (ë)	(137 * 27) mod 256 = 115 (s)	(115–31) mod 256 = 84 (T)	(84 XOR 23) = 67 (C)
165 (N))	(165 XOR 147) = 54 (6)	(54 + 17) mod 256 = 71 (G)	(71 * 27) mod 256 = 125 (})	(125–31) mod 256 = 94 ($\wedge$)	(94XOR 23) = 73 (I)
13 (♪)	(13 XOR 147) = 158 (R)	(158 + 17) mod 256 = 175 (>>)	(175 * 27) mod 256 = 117 (u)	(117–31) mod 256 = 86 (V)	(86 XOR 23) = 65 (A)
110 (n)	(110 XOR 147) = 253 (^{2})	(253 + 17) mod 256 = 14 (ψ)	(14 * 27) mod 256 = 122 (z)	(122–31) mod 256 = 91 ([)	(91 XOR 23) = 76 (L)

Original Plain text which was transmitted by sender was SOCIAL.

$$K = R^s modp = (((p-1).g)^{s.x})^r modp = ((p-1).g)^{s.x.r} modp \tag{3}$$

The receiver calculates the key K as,

$$K = S^r modp = (((p-1).g)^{r.x})^s modp = ((p-1).g)^{s.x.r} modp \tag{4}$$

The above two equations show the mathematical proof on how the sender and receiver receives the same key value ‘K’.

10. Generation of keys for pentaplicative cipher technique

The above method of Key sharing using the enhanced Diffie-Hellman Key Sharing algorithm feeds the five keys to the PentaPlicative Cipher Technique which then encrypts the message and sends the encrypted text to the receiver over the unsecure channel. For generating five keys, the Enhanced Diffie-Hellman needs to be called five times. For each iteration, the shared key ‘x’ is the current epoch timestamp. The keys ‘s’ and ‘r’ change for every iteration and are calculated as shown in the table below:

- Iteration 1: ‘s’ and ‘r’ are current latitude of both sender and receiver
- Iteration 2: ‘s’ and ‘r’ are current longitude of both sender and receiver
- Iteration 3: ‘s’ and ‘r’ are IP address of both sender and receiver
- Iteration 4: ‘s’ and ‘r’ are MAC address of both sender and receiver
- Iteration 5: ‘s’ and ‘r’ are fetched from randomly assigned numbers to both sender and receiver and these numbers change upon each time sender and receiver connect to the Base-Station.

Once these keys are generated, the sender and receiver can employ them to encrypt as well as decrypt the message which is being shared on both ends. The algorithm for key generation using enhanced Diffie-Hellman technique is described below:

- Step 1: start
- Step 2: load public keys ‘p’ and ‘g’
- Step 3: load private keys for sender ‘s’ and receiver ‘r’
- Step 4: obtain the shared session key ‘x’ from server
- Step 5: Compute sender key ‘S’ as ((p-1) * g)$^{s.x}$ mod p
- Step 6: Compute receiver key ‘R’ as ((p-1) * g)$^{r.x}$ mod p
- Step 7: exchange keys ‘S’ and ‘R’ amongst sender and receiver
- Step 8: Sender finds out actual key K_i as R^s mod p
- Step 9: Receiver finds out actual key K_i as S^r mod p
- Step 10: Repeat steps 3 to 9 for every iteration ‘i’ where i = 5
- Step 11: end

The algorithm briefed above is used to find out the keys K1, K2, K3, K4, and K5 which are then further used to encrypt the plain text message which the sender wants to send by making use of the PentaPlicative Cipher Technique which was discussed above. The same set of keys is also used on the receiver end to decrypt the cipher text received by the receiver by making use of the same PentaPlicative Cipher Technique’s decryption mechanism.

11. Enhanced Diffie-Hellman vs traditional Diffie-Hellman

The traditional Diffie-Hellman Key Exchange protocol suffers from a major drawback. There is no way for the sender and the receiver to validate and verify each other's identity. This poses a serious threat that a sniffer may just intercept the messages coming from sender and receiver and modify the messages by using his own key and then forward the modified messages to both sender and receiver. The sender and receiver can not tell if the message they received was from the authentic entity. Thus upon receiving the forged message from sniffer they perform the key operations on it and send the data back. The sniffer gets back two keys from both sender and receiver, which he can use to obtain all the messages being transferred between sender and receiver.

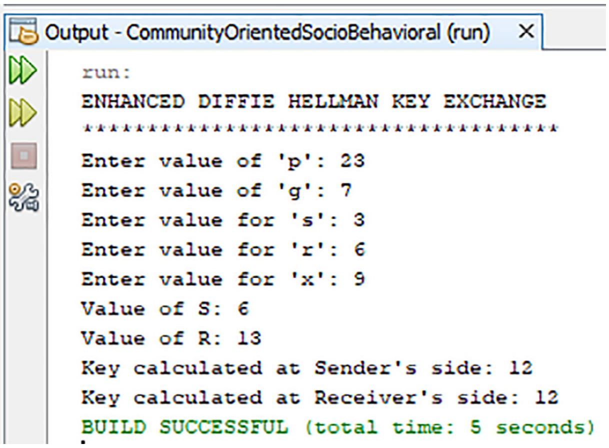
To overcome this pitfall, the enhanced Diffie-Hellman introduced a shared secret session key which will enable the sender and receiver to confirm their identity and also allow both of them to authenticate each other during the connection handshake. This shared session key is generated by the server and is also used during the key exchange and its key computation. A simple SHA256 HASH key can be used as a session key and on every iteration this key is regenerated to keep the crypto-system secure. Table 6 differentiates between the Enhanced Diffie-Hellman and Traditional Diffie-Hellman techniques by making use of a simple example (taken from [1]) followed by the simulation of both Enhanced Diffie-Hellman (Fig. 1) as well as the Traditional Diffie-Hellman techniques (Fig. 2).

12. Mathematical modeling

As stated above, the Community Based Socio-Behavioural Cipher technique is a combination of the PentaPlicative Cipher Technique with the Enhanced Diffie-Hellman Key Sharing algorithm to safely secure the community-based networks.

Table 6
Example to differentiate between Enhanced and Traditional Diffie-Hellman.

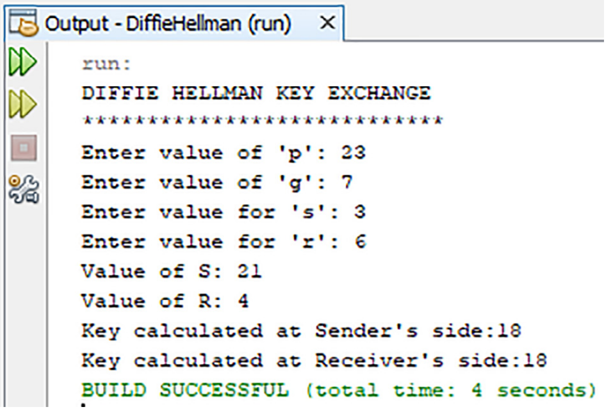
S. No.	Enhanced Diffie-Hellman	Traditional Diffie-Hellman
1	Choose public keys p = 23, g = 7	Choose public keys p = 23, g = 7
2	Choose sender key 's' = 3	Choose sender key 's' = 3
3	Choose receiver key 'r' = 6	Choose receiver key 'r' = 6
4	Choose session key 'x' = 9	–
5	S = $(7)^{(3*9)}$ mod 23 = 6	S = 7^3 mod 23 = 21
6	R = $(7 * (23 -1))^{(6*9)}$ mod 23 = 13	R = 7^6 mod 23 = 4
7	Ks = 13^3 mod 23 = 12	Ks = 4^3 mod 23 = 18
8	K_r = 6^6 mod 23 = 12	K_r = 21^6 mod 23 = 18



Output - CommunityOrientedSocioBehavioral (run)

```
run:
ENHANCED DIFFIE HELLMAN KEY EXCHANGE
************************************
Enter value of 'p': 23
Enter value of 'g': 7
Enter value for 's': 3
Enter value for 'r': 6
Enter value for 'x': 9
Value of S: 6
Value of R: 13
Key calculated at Sender's side: 12
Key calculated at Receiver's side: 12
BUILD SUCCESSFUL (total time: 5 seconds)
```

Fig. 1. Simulation of Enhanced Diffie-Hellman.



Output - DiffieHellman (run)

```
run:
DIFFIE HELLMAN KEY EXCHANGE
***************************
Enter value of 'p': 23
Enter value of 'g': 7
Enter value for 's': 3
Enter value for 'r': 6
Value of S: 21
Value of R: 4
Key calculated at Sender's side:18
Key calculated at Receiver's side:18
BUILD SUCCESSFUL (total time: 4 seconds)
```

Fig. 2. Simulation of Traditional Diffie-Hellman.

To perform mathematical operations on the character set which is input by the user, the PentaPlicative cipher technique encodes them into the ASCII values which are mapped easily from character to a number, on which it is easy to perform the set of mathematical operations. The encryption function given by, $T_n(y)$, is a resultant output of applying a set of pre-defined mathematical operations as cipher functions depicted by the function C(y). The mathematical modelling of the encryption mechanism can be depicted by employing aggregate mathematical equations that are explained in the text as follows:

C(y) = Bit_Dispersion(E5(y))
where, E5(y) = (E4(y) XOR K5(x)),
and, E4(y) = (E3(y) + K4(x)) mod 256,
and, E3(y) = (E2(y) * K3(x)) mod 256,
and, E2(y) = (E1(y) - K2(y)) mod 256,
and, E2(y) = ((P(y) XOR K1(y))
where P(y) defines the number of characters in the plain text = N_n

and, K(y) is the set of encryption keys which are obtained from the Enhanced Diffie-Hellman.

Key Sharing method that are further used in the encryption process as the five key functions:K1(y), K2(y), K3(y), K4(y) and K5(y). These key functions are used individually through all characters of plaint text with length N_n. Since the operation does not do one-to-one mapping with characters, hence the obtained length of cipher text C(y) is M_n^*

1. The generation of Keys K1(y), K2(y), K3(y), K4(y) and K5(y) are handled by the Enhanced Diffie-Hellman Key sharing algorithm. The Enhanced Diffie-Hellman uses the public keys 'p' and 'g' along with the private keys of the sender 's' and receiver 'r' along with the shared session key 'x' to generate and share the key function K(y).
2. The five keys are generated as part of five iterations of running the Enhanced-Diffie Hellman algorithm on different keys. Each iteration is a linear operation in nature and each linear operation execution contributes to the total running time of the cipher technique. The iterations are similar in nature with only exception being the choice of the keys 's' and 'r' for sender and receiver.
3. Upon successful application of the operations, the result obtained needs to be decoded from numbers back into the original ASCII character set from which it was earlier encoded. The functional integral values for C(y) is obtained individually as C0(y), C1(y), C2(y), C3(y)… Cn(y) for n length; n belonging to Natural number set 1, 2.. N and specified range $0 < C(y) < 255$. Conclusively, the numbers are

mapped back to the original character set via their ASCII code values that pertain to y_i as (y_0, y_1,...y_n) where the ASCII values are in Base10 order.

4. The integers thus obtained would first be converted into their respective Binary Base2 form and this Decimal to Binary conversion is carried out for all the values y_i by following the below mentioned procedure:
   Q0 = y_i/ 2(remainder value x_0)
   Q1 = y_0/ 2 (remainder value x_1)
   Q2 = 1/ 2 (remainder value x_2) and so on……..R_n (x_0,x_1,...x_7) until the quotient is 0, where n belongs to natural number Integer set I i.e. R_1 R_2...R_n.and this goes on to the number of characters in input plain text.
5. The obtained Base10 values for each individual character are then mapped into corresponding Binary values denoted as a function $R_n(y)$ and left shift operation follows:
   $L_n = \sum R_n(y + Ki)$ where the 0 < K_i < number of characters in plaintext, n.
   The encoded character set is treated with a special bit dispersion operation wherein, the number of bits of each character are mapped from eight bits to six bits of new character. The new characters are depicted as (l_0 l_1 l_2….l_{n^*}). h_0 ...h_5 denote the 6 bit character value of each integer.
6. The bit dispersion operation returns a 6-bit binary number which needs to be transformed into a decimal number of Base10. This Base2 to Base10 conversion makes use of the following transformation steps as depicted:
   W_n*= $h_0\times 2^0$ + $h_1\times 2^2$ +...+ $h_k\times 2^k$
   = h_0 + ($h_1\times 2$) + ($h_2\times 4$) +...+ ($h_k\times 2^k$), where (n^* belongs to the set of Integers, I) up till the length of plaintext.
   = h_0 + (2 × h_1) + (2 × ($h_2\times 2$)) +...+ (2 × ($h_k\times 2^{k-1}$))
   = h_0 + 2(h_1 + ($h_2\times 2$) +...+ ($h_k\times 2^{k-1}$)).
7. Conclusively, these transformed decimal numbers are then individually mapped into an ASCII character. The above equation shows each decimal number being represented as W_n*. Applying such mapping gives the final encrypted text and the function $T_n(y)$ gives the cipher text with length (n^*), which is greater than length of plain text.
8. The average execution time is given by equation, $T = (\Delta T_0 + \Delta T_1 + \Delta T_2 + \Delta T_3 + \Delta T_4 + \Delta T_5 + \Delta T_6 + \Delta T_7 + \Delta T_8 + \Delta T_9 + \Delta T_{10} + \Delta T_{11} + \Delta T_{12} + \Delta T_{13} + \Delta T_{14} + \Delta T_{15})/16$.
9. The Time complexity can be computed and depicted in Big-Oh notation as O (n) where the number of characters in the input Plain Text is depicted by n.
10. The calculation of the Time taken for various processes is specified in Table 7.

Table 7
Time Calculation.

S.No.	Operations	Time Taken
1	$K_1(y)$	$\triangle T_0$
2	$K_2(y)$	$\triangle T_1$
3	$K_3(y)$	$\triangle T_2$
4	$K_4(y)$	$\triangle T_3$
5	$K_5(y)$	$\triangle T_4$
6	$E_1(y)$	$\triangle T_5$
7	$E_2(y)$	$\triangle T_6$
8	$E_3(y)$	$\triangle T_7$
9	$E_4(y)$	$\triangle T_8$
10	$E_5(y)$	$\triangle T_9$
11	C(y)	$\triangle T_{10}$
12	ASCII Convert	$\triangle T_{11}$
13	Base10 to Base2	$\triangle T_{12}$
14	The bit dispersion	$\triangle T_{13}$
15	Base2 to Base8	$\triangle T_{14}$
16	Reverse ASCII	$\triangle T_{15}$

13. Results and discussions

The key generation operations using the enhanced Diffie-Hellman approach may seem costly with many operations being run, however, all the operations are linear. The single key generation test run of the Enhanced Diffie-Hellman code written in Java and executed on a Windows 10 machine running an Intel i5 processor with 8 GB RAM which hosted the JDK1.8 took an average of 1.3 ms. This run is for a single key generation and the same process would be repeated for five keys. The running time of PentaPlicative Cipher on the same machine (as mentioned above and full specifications briefed below) was calculated and the result reported in the original PentaPlicative Cipher Technique paper was **1.9 ms**. Hence, the total running time of securing the data by generating five different private keys and using them to encrypt the data before transmitting would be less than 10 ms. The below chart in Fig. 3 is a time comparison graph between the Key generation process cumulatively for five iterations to calculate the five keys and the time it takes to encrypt the data by using the PentaPlicative Cipher Technique. On average, the key generation process for a single key took 1.3 s as briefed above. Table 8 shows the specifications of the simulation environment which was used to code the Community Based Socio-Behavioural Cipher Technique. Fig. 4 draws a comparison of the Community Oriented Socio-Behavioural PentaPlicative Cipher Technique with the previously published technique [12] of the authors: Triplicative Cipher Technique. The runtime comparison has been made by using the same key sharing algorithm to share the keys for the techniques as well as to encrypt the data by using their respective cryptography model. The Triplicative technique uses three keys for encrypting the data and hence the key sharing time for Triplicative was less. However, the Community Oriented Socio-Behavioural PentaPlicative Cipher Technique uses five keys for encryption and hence key sharing time is greater and hence the overall time for technique is more than the Triplicative Cipher Technique.

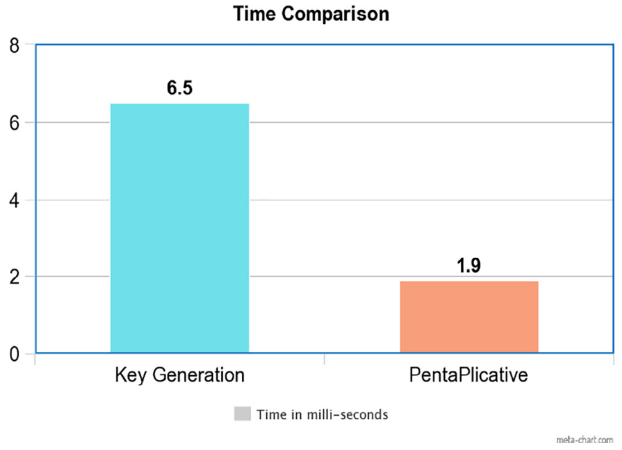


Fig. 3. Time Comparison between Key Generation and Pentaplicative Cipher technique.

Table 8
Simulation Environment Table.

O.S. used	Windows 10 Pro 64 bit
Processor	Intel Core i5 3230 M
RAM	8 GB
Simulation IDE	NetBeans
Version of IDE	8.2.0 build 201403101706
Development Language	Java
Java Version	1.8.0 build 25.25-b02

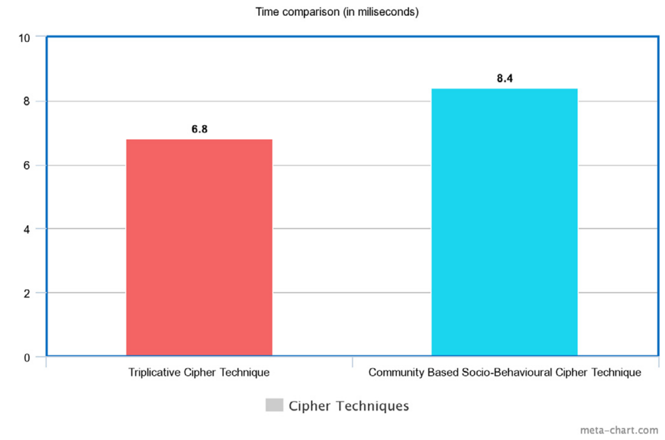


Fig. 4. Time Comparison between Triplicative Cipher Technique and Community Oriented Socio-Behavioural PentaPlicative Cipher Technique.

14. Conclusion and future work

The presented technique worked well on all devices, be it a mobile device or a personal computer or a remote host. However, efforts will be aimed in the future to reduce the battery consumption of the technique to optimize the performance of the technique without compromising on the security. This would enable the technique to be plugged into the mobile devices or remote hosts where the battery consumption is viable and be able to encrypt the data flowing out from these devices. To achieve such battery optimization, the caching of keys by the devices to stop regeneration at every iteration and lowering the number of pings by background service to base station may be some of the approaches to start with. Similarly, future work would be focused on to improve the performance of the current approach presented here. Another future addition to the technique here is the introduction of a single unique identifier as a single key which would eliminate the need of random keys to be chosen by the sender and receiver as their private key 's' and 'r'. This unique identifier comprise a set of information. This set of information combined would be unique to every individual and can be mapped to every single individual in the world. The composition of this unique identifier is briefed below as a set of 3 characters each:

```
---   ---   ---      ---   ---   ---   ---
      ---   ---
CnC   CC    SC       DC    TC    VC    FC
      PC    Parity
```

Here, the CnC is Continent Code, CC is Country Code, SC denotes State Code, DC is District Code, TC denotes Tehsil Code, VC signifies Village Code, FC is the Family Code (person's family tree can have a unique code based on their last name), PC is the Person Code and the last 3 characters denote the Parity bits which are assigned in a random fashion to every individual to ensure the uniqueness of this entire set of 27 bits. This unique identifier along with the latitude and longitude coordinates of current location of sender as well as receiver and the timestamp would make up for the private key set. Since, at least the timestamp varies on every iteration, hence the set of keys being used for Diffie Hellman key exchange mechanism are always different and this caters to the security as pointed earlier without relying on generating random numbers. Further efforts in the future would be made to polish and work on this rough proposal of using the Unique Identifier and generating the keys out of this.

Funding

The funding of the article that involves payment of APC (Article Processing Charges) towards the publication of the research paper would be borne by the author(s) from their internal resources.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgement

The author(s) would like to thank administration of Guru Gobind Singh Indraprastha University, Dwarka, Delhi-110078 (INDIA), University of Delhi, JNU Delhi for providing research oriented academic environment.

References

[1] Forouzan BA, Mukhopadhyay D. Cryptography and network security, vol. 12. USA: Mc Graw Hill Education (India) Private Limited New York, NY; 2015.

[2] W. Stallings, Cryptography and network security principles and practices 4th edition; 2006.

[3] N.A. Azeez, C.V. der Vyver, Security and privacy issues in e-health cloud-based system: A comprehensive content analysis, Egypt Inform J 21, vol. 02, pp. 97–108, 2019.

[4] F.R. Shareef, A novel crypto technique based ciphertext shifting, Egypt Inform J 21, vol. 02, pp. 83–90, 2020.

[5] A.K. Bairwa, S. Joshi, Mutual authentication of nodes using session token with fingerprint and mac address validation, Egypt Inform J 22, vol. 04, pp. 479–491, 2021.

[6] A.G.R.H.A.F. Sayed, Wafaa S., A. Elsedeek, Trajectory control and image encryption using affine transformation of lorenz system, Egypt Inform J 22, vol. 02, pp. 155–166, 2021.

[7] A.F. Khan, G. Anandhara, Ahkm: an improved class of hash based key management mechanism with combined solution for single hop and multi hop nodes in iot, Egypt Inform J 22, vol. 02, pp. 119–124, 2021.

[8] F. Kausar, Iris based cancelable biometric cryptosystem for secure healthcare smart card, Egypt Inform J 22, vol. 04, pp. 447–453, 2021.

[9] I. Keshta, A. Odeh, Security and privacy of electronic health records: Concerns and challenges, Egypt Inform J 22, vol. 02, pp. 177–183, 2021.

[10] Li Q, Zhang L, Zeng F, Pan Y, Yang J. Community clustering routing algorithm based on information entropy in mobile opportunity network. IEEE Access 2022;10:25755–66.

[11] Li S, Zhu B, Zhu H, Liu F, Zhang Y, Wang R, Lu H. Heterogeneous attention concentration link prediction algorithm for attracting customer flow in online brand community. IEEE Access 2022;10:20898–912.

[12] Rana R, Berg K, Degefa MZ, Löschenbrand M. Modelling and simulation approaches for local energy community integrated distribution networks. IEEE Access 2022.

[13] Sahraoui Y, De Lucia L, Vegni AM, Kerrache CA, Amadeo M, Korichi A. Traceme: Real-time contact tracing and early prevention of covid-19 based on online social networks. In: 2022 IEEE 19th Annual Consumer Communications & Networking Conference (CCNC). IEEE; 2022. p. 893–6.

[14] Jastania Z, Abbasi RA, Aslam MA, Khanzada TJS, Ghori KM. Analyzing public discussions about# saudiwomencandrive using network science. IEEE Access 2022.

[15] Xu X, Tang J, Xiang H. Data transmission reliability analysis of wireless sensor networks for social network optimization. J Sens 2022;2022.

[16] Wanda P. Modern privacy-preserving and security schemes in social networks: A review. Int J Inform Comput 2022;3(2):23–40.

[17] Xiao Y, Jia Y, Cheng X, Wang S, Mao J, Liang Z. I know your social network accounts: A novel attack architecture for device-identity association. IEEE Trans Dependable Secure Comput 2022.

[18] Dang-Pham D, Kautz K, Hoang A-P, Pittayachawan S. Identifying information security opinion leaders in organizations: Insights from the theory of social power bases and social network analysis. Comput Secur 2022;112: 102505.

[19] N.C.S.D.A.N.A.N.D. Sharma, Dilip Kumar, J. Sivakumar, A review on various cryptographic techniques & algorithms, Mater Today: Proc 51, pp. 104–109, 2022.

[20] F. Artuğer, F. Özkaynak, A method for generation of substitution box based on random selection, Egypt Inform J 23, vol. 01, pp. 127–135, 2022.

[21] Feng J, Li L. Scenery: a lightweight block cipher based on feistel structure. Frontiers of Computer Science; 2022. p. 1–10.

[22] X.X.S.J.W.G.D.G.H.Z.S.M.Z.Q. Jin, Xuancheng, J. Li, Annotating, tracking, and protecting cryptographic secrets with cryptompk, IEEE Symposium on Security and Privacy (S&P), 2022.

[23] A.C.G.S.C.R.B.K.V. William, P., S. Choubey., Assessment of hybrid cryptographic algorithm for secure sharing of textual and pictorial content, International Conference on Electronics and Renewable Systems (ICEARS), pp. 918–922, 2022.

[24] Garg N, Bhatia H, Johari R. Pentaplicative cipher technique. In: International Conference on Innovative Computing and Communications. Springer; 2019. p. 241–9.

[25] Johari R, Bhatia H, Singh S, Chauhan M. Triplicative cipher technique. Proc Comput Sci 2016;78:217–23.

Siya Garg Experienced Teacher with a demonstrated history of working in the Education industry. Skilled in Visual Basic, C++, Java, HTML, CSS,Python and Java script.Strong education professional with a Master's Degree focused in Informatics from Institute of Informatics and Communication, University of Delhi. She is currently affiliated with Department of Computer Science, Keshav Mahavidyalaya, University of Delhi.

Vinita Jindal She has done her graduation in Mathematics from the University of Delhi, MCA (Masters in Computer Application) from Indira Gandhi National Open University. Then she pursued M.Phil. (Computer Science) from Madurai Kamaraj University followed by a Ph.D. (Computer Science) from the University of Delhi. Currently, she is employed as an "Associate Professor" in the Department of Computer Science, Keshav Mahavidyalaya, University of Delhi.

Harshit Bhatia Experienced Web and Mobile Application Developer with a demonstrated history of working in the computer software industry. Having a knack for research in cryptography demonstrated by multiple publications in international conferences and journals and a patent. Strong information technology professional with a Master of Computer Applications (M.C.A.) focused in Computer Science from Guru Gobind Singh Indraprastha University.

Rahul Johari Dr. Rahul Johari is teaching at University School of Information and Communication Technology (USICT), Guru Gobind Singh Indraprastha University, Dwarka, Delhi, India. Presently, he is the Head of the Software Development Cell and Head and Founder of SWINGER [Security, Wireless, IoT Network Group of Engineering and Research] Lab.

Shrey Gupta Shrey Gupta is currently a student in fourth year pursuing his B.Tech. (Bachelor in Technology) in Electronics and Communication Engineering from University School of Information Communication and Technology (USICT), Guru Gobind Singh Indraprastha University, Dwarka, Delhi, India. He is currently affiliated with SWINGER [Security, Wireless, IoT Network Group of Engineering and Research] Lab.