International Journal of Computer Networks and Applications (IJCNA)
DOI: 10.22247/ijcna/2024/46 Volume 11, Issue 6, November – December (2024)

RESEARCH ARTICLE

A Robust Security Strategy Using Hybrid Cryptography Approach to Protect Data in the Financial Sector

Bilas Haldar
Department of Computer Science and Engineering, The Neotia University, Sarisha, South 24 Parganas, West Bengal, India.
bilasphd2020@gmail.com

Partha Kumar Mukherjee
Department of Computer Science and Engineering, The Neotia University, Sarisha, South 24 Parganas, West Bengal, India.
✉ parthakumar.mukherjee@tnu.in

Himadri Nath Saha
Department of Computer Science, SNEC, Calcutta University, Kolkata, West Bengal, India.
contactathimadri@gmail.com

Received: 12 June 2024 / Revised: 13 November 2024 / Accepted: 04 December 2024 / Published: 30 December 2024

Abstract – Ensuring data security has become an imperative aspect across various sectors particularly with the sensitivity of the financial sector. Electronic Banking (E-Banking) serves as a fundamental conduit for accessing information about banks and their services through the Internet. The security of financial data is crucial for building and maintaining customer trust. However, emerging threats are gradually increasing to breach the data in the financial sector. The developed methodology fails to fully protect data against increasingly sophisticated cyberattacks with higher efficiency of the performance. To address these challenges, this work proposed an innovative hybrid cryptography approach designed to strengthen digital data protection in the financial sector. The hybrid cryptography technique combined with proposed Modified IDEA (M-IDEA) with a 512-bit key size, and a Double Secure RSA (DS-RSA) methodology to enhance data protection in the financial sector. Additionally, this work introduces a novel method for generating keys in multi-party authentication systems. The work presents the innovative encryption and decryption methodology for encrypting and decrypting authentication keys and data files. The performance of the suggested hybrid cryptography methodology compared with the traditional methodology showcases its potential advancements in data security. The outcome of the proposed key generation technique varies from 0.60656 to 3.83477 seconds of the prime numbers 2 to 10. Furthermore, the encryption process time fluctuates between 0.62702 and 19.69348 seconds with file sizes of 0.12 to 3.68 MB. Moreover, this research work extends the security defense strategy to address vulnerabilities related to sensitive data disclosure breaches and man-in-the-middle attacks on applications within the financial sector. Consequently, these advancements play a crucial role in bolstering the integrity of digital interactions and transactions in the domain of electronic banking.

Index Terms – Key Generation, Encryption, Decryption, M-IDEA, DS-RSA, Man-in-the-Middle Attack, Sensitive Data Disclosure Breaches Attack, Security Strategy Framework.

1. INTRODUCTION

Security has become a paramount concern across all facets of life in the digital world. The broad adoption of computers and internet usage for communication illustrates the critical importance of data security in facilitating secure interactions and the protected transfer of information over connected networks. The financial sector has transformed the way consumers interact with banking institutions, offering unprecedented convenience in accessing banking services through the Internet. However, the financial sector is increasingly attacked by sophisticated cyberattacks, ranging from data breaches to malware attacks. It threatens the confidentiality, integrity, and availability of financial transactions and customer data. As the financial sector embraces digital technologies, ensuring the security of online transactions, customer information, and financial data is critical to preserving trust and maintaining system integrity. The development of robust safety methodology inside the financial industry is necessary to prevent potential consumer attacks, particularly in banks. The erosion of trust resulting from security concerns can significantly impact on the adoption of E-banking systems. Traditional security approaches are increasingly vulnerable to sophisticated cyber threats, compelling researchers to explore advanced techniques in biometrics, cryptography, and data embedding to safeguard banking systems. The subkey generation technique using the IDEA algorithm is introduced to ensure the security of the methodology [1]. The novel approaches to RSA encryption, are aimed at mitigating risks of cyber security attacks [2] and strengthening the encryption algorithms' security [3]. The innovative approach to physical-layer security that encapsulates cutting-edge encryption technology is new horizons in secure communications [4]. The vulnerabilities of financial data security on the RSA algorithm are due to small key sizes and the reliance on prime number values [5][6]. The IDEA and an RSA algorithm variant using gaussian interpolation increased algorithm complexity and susceptibility to supply chain attacks that undermine the integrity of cryptographic solutions [7] [8].

The App-based methodology was designed to detect vulnerable implementations of One-Time Password (OTP), Short Message Service (SMS) APIs in the banking sector [9]. Mobile-based user authentication is another area of concern, as many financial institutions rely on these methods to facilitate secure transactions. The importance of transforming authentication technologies to enhances the security for mobile money transactions, which are increasingly popular but can be vulnerable despite existing legal and procedural safeguards [10]. The impact of cyberattacks on financial institutions and highlighted the potential global economic consequences. It provides a stark reminder of the high stakes involved in financial cybersecurity and the necessity for advanced protection mechanisms to defend against increasingly sophisticated cyber threats [11]. An innovative Near Field Communication (NFC)-based authentication protocol specifically for mobile payments is introduced to address user privacy and anonymity concerns. The approach exemplifies the drive for advancements in mobile payment security that also meet user expectations for convenience and confidentiality [12]. The potential of biometric authentication mitigates the risks of cybersecurity, underscoring its value in enhancing user safety and data integrity within financial institutions that explained in [13]. Cybersecurity measures are essential for protecting information and managing risk, as demonstrated by [14], who illustrated the significant impact of cybersecurity practices in the banking sector. An innovative steganography approach that combines the International Data Encryption Standard Algorithm with the Least Significant Bit Grouping (LSBG) method is used to address emerging data protection challenges [15]. A dynamic One-Time Password (OTP) model integrated with the RSA algorithm that minimizes vulnerabilities associated with static encryption keys, providing a novel direction for secure transaction verification [16]. The role of multifactor authentication as an essential layer in the security framework, aimed at reinforcing transaction security and mitigating risks associated with unauthorized access. However, despite the adoption of these technologies, challenges remain particularly in balancing security and user convenience, as well as adapting to increasingly sophisticated cyber threats [17][18].

Despite the development of various security mechanisms, traditional cryptography methods have shown limitations in addressing emerging cyber threats, particularly in terms of performance efficiency and the protection against advanced attack vectors. Current cryptography techniques, while effective in some scenarios, often fall short in providing the necessary level of security and speed required by modern E-Banking systems. The increasing complexity of attacks has exposed the vulnerabilities in existing solutions, creating an urgent need for more robust and efficient approaches to protect financial data. To address these challenges, this work proposed an innovative hybrid cryptography approach that combines Modified IDEA (M-IDEA) and Double Secure RSA (DS-RSA) techniques. The suggested hybrid cryptography approach aims to enhance data security and offer a robust efficient solution for the protection of sensitive financial data in the E-Banking sector. The key size of the M-IDEA algorithm is 512-bit. This key is used for the encryption and decryption process during the transmission of keys and data files. The DS-RSA algorithm uses a random function to generate the public and private keys in key generation methodology that increases the hardness of the key.

Initially, data is converted from readable to unreadable using the M-IDEA with the key size of a 512-bit technique. After that unreadable data is again encrypted using the DS-RSA algorithm. It is very difficult to break the system as it is encrypted two times using the proposed hybrid cryptography methodology. It is harder to breach the data for the attacker as there are lots of mathematical computations added within the proposed methodology. During the decryption process, data is initially decrypted using the DS-RSA algorithm. Then data is again decrypted using M-IDEA with 512-bit methodology. This double encryption and decryption approach increases the complexity of the proposed methodology, and it is not easy to break it. The main goal of this research work is to develop a system that can encrypt and decrypt various types of multimedia data, including text, images, PDFs, executables, videos, and audio files using a proposed hybrid cryptography methodology. This innovative approach is anticipated to contribute significantly to enhancing the security of the banking environment, catering to both customers and bankers through improved authentication processes. The result of the proposed methodology shows improvement in performance concerning security. The research findings underscore the efficacy of the proposed algorithm, revealing enhanced speed and security for online transactions. Furthermore, this work introduces a security defense framework employed within the financial sector to address and prevent attacks related to the disclosure of sensitive data breaches. The proposed security defense strategy in the financial sector refers to a comprehensive and systematic technique designed to protect information and communication systems from unauthorized access attacks, and potential breaches.

1.1. Objectives

The primary objective of this research work is to address the identified research gap by developing and evaluating an innovative, reliable, and effective key generation method specifically tailored for financial organizations. The overarching goal is to contribute to the advancement of universal encryption frameworks capable of seamless integration across diverse domains, with a particular focus on enhancing cybersecurity measures in electronic banking systems. The specific objectives are as follows:

Design and implement a universal encryption and decryption technique that leverages the strengths of Modified IDEA (M-IDEA) and Double Secure RSA (DS-RSA) algorithms

Introduce an innovative and reliable key-generation method tailored for the financial sector, addressing the specific security requirements and challenges inherent in financial transactions.

Conduct a comprehensive performance analysis of the proposed hybrid cryptography approach against traditional and current encryption methods.

To enhance data security in the financial sector by developing an improved hybrid cryptography technique that integrates M-IDEA with a 512-bit key size and DS-RSA methodology.

Contribute to the development of more robust, efficient, and versatile cybersecurity solutions specifically tailored for financial organizations.

To extend the security defense strategy to combat vulnerabilities associated with sensitive data disclosure breaches and application-level attacks in the financial sector

1.2. The Principal Contributions of the Work are as Follows

The work introduces a novel approach by combining M-IDEA with a 512-bit key size and DS-RSA algorithms to enhance the security of data transmission.

Development of a novel method for generating keys in a multi-party authentication technique that addresses the need for secure key management in complex authentication scenarios.

The work presents an innovative hybrid cryptography approach for the encryption and decryption of authentication keys and data files using M-IDEA and DS-RSA algorithms. This contributes to enhancing the overall security of data transmission within the financial sector.

This research evaluates the performance of the newly suggested hybrid cryptography method in comparison to the traditional RSA algorithm. This includes an analysis of factors such as encryption and decryption speed, key generation efficiency, and overall system performance.

The proposed methodology is implemented within the financial sector to ensure the secure transformation of data.

The approach introduces a security strategy framework within the financial sector, aimed at safeguarding confidential data.

The work extends the strategy of sensitive data disclosure breach attacks to address vulnerabilities within the financial sector. This comprehensive approach to security contributes to the development of more robust defense mechanisms against a range of cyber threats.

1.3. The Paper is Organized as Follows

Section 2 provides an extensive review of relevant literature, establishing the foundation and context for the current research. Section 3 presents proposed methodology, providing a clear exposition. The section 4 showcases the results obtained from simulations and includes an in-depth discussion of the effectiveness and implications of the proposed methodology. Analyzes the performance metrics, including a comparative evaluation of the obtained results is introduces in Section 5. Furthermore, Section 6 explores the security aspects of the methodology and demonstrates its practical applicability within the financial sector. Finally, Section 7 concludes the research work with thoughtful observations and conclusions.

2. LITERATURE REVIEW

Recent literature underscores a concerted effort toward developing sophisticated key generation, encryption, and decryption methodologies to safeguard digital transactions and communication in the evolving cybersecurity landscape. This literature review examines several research studies that contribute to understanding cybersecurity in the banking sector. It focuses on detecting vulnerable implementations of OTP SMS, mobile-based user authentication, the impact of cyberattacks on financial institutions, cybersecurity techniques based on hybrid cryptography, artificial intelligence, and authentication protocols for online banking. Mishra et al. [19] presented an artificial intelligence-driven cybersecurity technique tailored for managing risks in the financial sector. The work likely discusses the application of AI in detecting and preventing cyber threats, enhancing risk management, and protecting financial systems. This method utilizes the K-Nearest Neighbor (KNN) algorithm in conjunction with an Enhanced Encryption Standard (EES) for robust encryption and decryption techniques. The objective of the methodology is to improve cyber defenses and enhance risk management across financial systems. The work reports performance gains in cybersecurity through AI-powered detection and prevention of attacks. However, limitations include a small sample size and a narrowly focused analysis, with minimal discussion of the practical implications and potential costs of the solution. Shivaramakrishna et al. [20] introduced an innovative hybrid cryptographic framework combining RSA and AES algorithms to enhance secure data storage in cloud computing environments. Their extensive performance evaluation reported impressive results, with accuracy, precision, recall, and F1-score values reaching 99.12%, 98.78%, 98.11%, and 98.56%, respectively. These metrics underscore the framework's effectiveness in secure data storage applications. Further research directions include optimizing computational efficiency, exploring scalability for large cloud infrastructures.

Abid et al. [21] addressed the computational challenges of encryption by developing an optimized Homomorphic Encryption algorithm, HE-CRT-RSA, which integrates the Chinese Remainder Theorem with the RSA framework. This approach resulted in enhanced performance, particularly through reduced decryption times, demonstrating potential efficiency gains for secure data processing. However, the work did not explain the practical implementation of Advanced Homomorphic CRT-RSA, particularly regarding its applicability in cloud environments and resilience to security threats. Stanikzai et al. [22] evaluated the effectiveness of existing cybersecurity methods in reducing and mitigating financial crime in the banking sector. The work provides a comprehensive review of information-related cyber-attacks, associated threats, major challenges, and control measures. The authors also examine specific cyber-attacks and present solutions to counteract these threats, offering insights into effective methods for reducing cybersecurity risks in financial institutions. Ghelani et al. [23] introduced an innovative banking system model that leverages biometric authentication and digital signatures to ensure secure transactions for bank customers. This model enhances security by leveraging the uniqueness of biometric identifiers and the non-repudiation properties of digital signatures. To enhance the model’s resilience against emerging threats and improve its usability for a broader range of users, it ensures both security and user-friendliness in banking applications. Sekhar et al. [24] provided insights into cyber attacks targeting the banking sector and proposed effective cyber security strategies to counter such threats. The authors highlight the vulnerabilities associated with cyber attacks in the banking industry and emphasize the need for proactive security measures.

Ganavi et al. [25] presented a method for enhancing image security through a two-level scrambling process for input secret images. This approach employs Rivest Cipher 6 (RC6) and One-Time Pad (OTP) algorithm to secure images against unauthorized access. Simulation results indicate that this method provides robust two-layer protection for color image transmission and key authentication. The approach makes it challenging for intruders to extract or identify the keys by using unique keys for each algorithm. Additionally, these keys transmit securely to recipients using the Modified Rivest-Shamir-Adleman (MRSA) algorithm, further strengthening the method’s security. The work done by Kanda et al. [26] is a proficient hardware implementation of the IDEA cipher, centering around using arithmetic modulo multiplication an essential computation in the IDEA algorithm. This implementation harnesses an innovative Vedic multiplier architecture for enhanced efficiency. Neela et al. [27] developed a secure cloud storage system in which data encryption is conducted by the data owner using an Improved RSA algorithm (IREA) before uploading to cloud services. This system eliminates the need for third-party involvement by combining IREA with the Flexible Capacity Cuckoo Filter (FCCF) auditing technique, ensuring data integrity without relying on external audits. Kuppuswamy et al. [28] introduced a hybrid encryption system that combines RSA and a Simple Symmetric Key (SSK) algorithm to secure financial transactions. This hybrid approach aims to enhance encryption strength and privacy by leveraging the benefits of both asymmetric and symmetric encryption. The work suggests that this hybrid algorithm offers improved security for financial data.

Maria et al. [29] developed a lightweight Elliptical Curve Cryptographic (ECC) methodology designed specifically for financial transactions conducted through mobile applications. The approach extends device longevity and mitigates overheating issues in mobile phones by reducing power consumption by 3.5%. However, future work is needed to minimize computational delay, potentially by increasing the input bit size to improve processing efficiency. Iyera et al. [30] presented a hybrid cryptography technique using AES and ECC algorithm. This combination enhances security by optimizing the key size-to-time ratio and improving the overall efficiency of the encryption process. Mangalagowri et al. [31] made significant contributions to cloud performance by developing an Enhanced Particle Swarm Optimization (EPSO) algorithm integrated with a Hypervisor Attack Detection framework. The experimental results demonstrate that this approach outperforms existing methods, offering improved reliability, higher throughput, and lower energy consumption. Hermawan et al. [32] introduced the EPNR (Eight Prime Numbers of Modified RSA) technique, a modified double RSA that incorporates eight prime numbers in conjunction with the Chinese Remainder Theorem (CRT) method. The primary objective of this approach is to enhance the speed of both random key generation and the decryption mechanisms. Wardak et al. [33] developed a novel cryptography mechanism aimed at securing data transmission and retrieval. Their approach utilizes a signed graph alongside its adjacency matrix and the RSA algorithm, offering a new perspective on data encryption methodologies. Karim et al. [34] presented a remarkably efficient security system centered around digital signature authentication. It is leveraging an asymmetric key cryptography algorithm in conjunction with token-based encryption. This system focuses on enhancing the security of digital transactions by ensuring the authenticity and integrity of transaction data.

Kaur et al. [35] introduced an innovative security mechanism involving DNA-based block chaining to prevent collisions and provide high-end security for financial transactions. This novel approach uses the unique properties of DNA sequences to create a secure and tamper-evident chain of transaction records. Merkepci et al. [36] presented a methodology to construct the neutrosophic version of the RSA crypto algorithm. This approach involves incorporating neutrosophic logic, which enhances the algorithm's capability to handle uncertainty and vagueness. However, their numerical analysis revealed that this modified version has approximately double the complexity compared to the classical RSA algorithm. Alatawi et al. [37] introduced a novel hybrid encryption algorithm combining AES, RSA, and Blowfish. The experimental results of the work demonstrate that this integrated solution successfully safeguards private data without compromising scalability. Additionally, the approach utilizes Logi-XGB as a predictive model to detect potential attacks, successfully blocking 99.7% of threats. Joseph et al. [38] developed an innovative Hybrid Bat and Cuckoo-based Pallier Homomorphic Encryption (HBC-PHE) scheme, which employs bat and cuckoo fitness functions within the Pallier Homomorphic Encryption framework. This hybrid approach is specifically designed to enhance data security, providing robust defense against malware and other types of attacks, making it a promising solution for secure data handling in sensitive environments. Alatawi et al. [39] introduced a Hybrid and Adaptive Cryptography (HAC) approach tailored for authentication on the Internet of Things (IoT). This approach aims to address security challenges in IoT environments by combining multiple cryptographic techniques. The proposed method utilizes the XOR operation, a hashing function, and a hybrid encryption strategy that integrates the RSA and AES algorithms. This combination enhances both the security and adaptability of authentication processes, making it well-suited for the dynamic and resource-constrained nature of IoT systems. Table 1 provides a summary of the key findings from the literature review.

Table 1 Summary of the Literature Review

Author	Methodology used	Limitations and future work
Mishra et al. [19]	KNN and EES algorithm	The research is limited to a small sample size and a narrow focus. Additionally, it lacks a detailed analysis of the potential costs and consequences of the proposed solution.
Shivaramakrishna et al. [20]	Integrating AES-OTP and RSA	Future work needs to focus on optimizing computing efficiency, scalability in large-scale cloud environments, and enhancing key management procedures for long-term data storage solutions.
Abid et al. [21]	Homomorphic encryption algorithm based on the Chinese Remainder Theorem and RSA.	The practical implementation of the Advanced Homomorphic CRT-RSA algorithm and the handling of cloud-specific security threats remain unexplored in future work.
Stanikzai et al. [22]	Analyzes the security of related information, cybercrime and cyber-attacks.	The methodology lacks a clear framework for preventing cyberattacks in the financial sector.
Ghelani et al. [23]	Biometric impressions and digital signatures	Future research will focus on improving security against unforeseen threats and enhancing user-friendliness.
Sekhar et al. [24]	Review the cyberattacks in the banking sector	Does not mention specific methodologies for protecting financial data, leaving gaps in practical application for security improvements.

Ganavi et al. [25]	Modified Rivest-Shamir-Adleman (MRSA)	The methodology lacks clarity regarding its ability to withstand brute force and other cryptographic or steganographic attacks.
Kanda et al. [26]	IDEA algorithm	Future work will focus on implementing the decryption core to create a unified IDEA crypto core, as the current decryption process is computationally intensive for hardware.
Neela et al. [27]	IREA along with the FCCF auditing technique.	Future research will focus on enhancing security to prevent third-party attacks, but the current study lacks a comprehensive security framework.
Kuppuswamy et al. [28]	RSA and SSK algorithm	The proposed method is efficient in encryption and decryption times but lacks a detailed security analysis.
Maria et al. [29]	Lightweight Elliptical Curve Cryptography	The method requires reducing computational delay by increasing the number of input bits.
Iyera et al. [30]	AES and ECC	Security analysis and results of the proposed methodology were not provided.
Mangalagowri et al. [31]	EPSO integrated with a Hypervisor Attack Detection framework.	Future work focus on developing effective soft computing and machine learning algorithms to address various prominent attacks.
Hermawan et al. [32]	RSA algorithm using eight prime numbers	The computation time increases significantly when using very large prime numbers.
Wardak et al. [33]	Signed graph, its adjacency matrix and the RSA algorithm	The proposed methodology requires additional security improvements.
Karim et al. [34]	RSA and hash function	The approach remains vulnerable to sensitive data breaches.
Kaur et al. [35]	DNA-based block chaining	The algorithm's complexity increases, making it more computationally intensive.
Merkepci et al. [36]	Neutrosophic version of the RSA crypto algorithm.	The method exhibits double the complexity compared to the classical RSA version.
Alatawi et al. [37]	AES, RSA, and Blowfish algorithms	The work does not address the potential risks of real-time attacks.
Joseph et al. [38]	HBC-PHE scheme	Requires robust security analysis to validate its effectiveness
Alatawi et al. [39]	XOR operation, a hashing function, and a hybrid encryption strategy based on the RSA and the AES algorithms.	The encrypted files are two to three times the size of the original files, which requires substantial additional storage space. Further research is needed to optimize this issue.

The literature reviewed studies cover a range of security measures and techniques for financial transactions, there is a potential research gap in exploring the integration of emerging hybrid cryptography technologies to further strengthen security in the banking sector. The potential research gap investigating the integration of advanced cybersecurity techniques with existing banking systems to enhance overall security and resilience. The literature indicates a variety of specialized encryption and decryption methods tailored for specific applications of secure data transmission in the financial sector. However, there appears to be a gap in developing universal encryption and decryption methodology that can seamlessly integrate with existing systems across different domains, facilitating broader adoption and standardization in encryption practices. Research is needed to develop adaptive encryption and decryption algorithms that can automatically adjust their security parameters in response to evolving threats and attack vectors.

A research gap exists in terms of understanding the practical challenges associated with implementing these techniques in real-world scenarios, especially in electronic banking systems. Investigating the practical viability, resource requirements, and potential obstacles to integration would contribute to a more comprehensive understanding of the applicability of these hybrid cryptographic methods. This proposed work fills this gap by introducing an innovative, reliable, and effective key generation method for financial organizations. The work can contribute to the development of robust, efficient, and versatile cybersecurity solutions capable of withstanding the challenges posed by the rapidly evolving landscape of cyber threats.

The method makes use of M-IDEA and DS-RSA algorithms for multi-party communications in financial organization. Furthermore, the research work presents a novel encryption, decryption, and security defense strategy technique that is used for secure data transformation over the network in the financial using the proposed hybrid cryptography approach.

3. PROPOSED METHODOLOGY

The proposed methodology for this research work employs a hybrid encryption system that combines symmetric encryption through M-IDEA and asymmetric encryption using the DS-RSA algorithm. Its foundations include the resilience of modular arithmetic, bitwise XOR, greatest common divisor, Euler totient function, prime number, and discrete logarithm methods. This work initially describes the designing and implementation of secure symmetric encryption and decryption techniques using M-IDEA methodology.

The key size has been augmented from 128 to 512 bits for enhanced security of the proposed methodology. Additionally, the methodology introduces DS-RSA algorithm for key generation, encryption and decryption techniques. This methodology aims to ensure the security of online financial transactions. It also provided robust security in multimedia data types, including text documents, images, audio, and video for transformation. Algorithm 1 outlines the proposed key generation process based on the DS-RSA methodology.

```
Require: Public key component (PK, FK, RN), Private key component (QK, GK, RN ), and Random number (RN ).
Ensure: 'n' random unique prime numbers a1, a2, a3, a4, a5, ……an
Select 'n' even random distinct prime numbers a1, a2, a3, a4, a5, ……an
Calculate value of x1, x2, x3, x4, …. xxn
$x1 = a1 × a2$
$x2 = a3 × a4$
$x3 = a5 × a6$
$xn = a7 × an$
$xm = x1× x2× x3× x4$
Calculate Euler ϕ() of x1, x2, x3, x4, ….. xn and xm
$ϕ(x1) = (a1 – 1) × (a2 – 1)$
$ϕ(x2) = (a3 – 1) × (a4 – 1)$
$ϕ(x3) = (a5 – 1) × (a6 – 1)$
$ϕ(xn) = (a7 – 1) × (an – 1)$
$ϕ(xm) = ϕ(x1) × ϕ(x2) × ϕ(x3) × ϕ(xn)$
Calculate value of x12, and x3n
$ϕ(x12) = ϕ(x1) × ϕ(x2)$
$ϕ(x3n) = ϕ(x3) × ϕ(xn)$
Create a random number, denoted as PK1, through a randomization process
GCD (PK1, ϕ(x12)) = 1 and 1 < PK1 < ϕ(x12)
Generate a random number PK2 € GCD (PK2, ϕ(x3n)) = 1 and 1<PK2<ϕ(x3n)
Calculate PK12 = (PK1)PK2 % xm
Generate a public key PK € GCD(PK, ϕ(xm) ×PK12) = 1 and (1 < PK <(ϕ(xm) ×PK12)
Generate a FK € GCD(FK, ϕ(xm)) = 1 and 1 < FK < ϕ(xm)
Calculate the private QK, € QK = PK−1% (ϕ(xm) × PK12)
Generate a GK € GCD(GK, ϕ(xm)) = 1 and 1 < GK < ϕ(xm)
Compute a random number RN €
if a1 > a2 then
    Satisfy (x1 - a1) < RN < x1
    GCD (RN, x1) = 1
    else
        if a1 < a2 then
            Satisfy (x1 - a2) < RN < x1
            GCD(RN, x1) = 1
        end if
    end if
Return RN
Exit
```

Algorithm 1 Key Generation Using DS-RSA

The key generation methodology presents in Algorithm 1 which ensures the keys are distinct, large, and unpredictable. These are essential properties for secure encryption and decryption techniques. The process involves generating coprime values with respect to Euler’s totient function of selected prime numbers. It ensures the keys which are valid for use in encryption and decryption techniques. The algorithm begins with the selection of distinct even prime numbers. These primes are combined to form large numbers that make it computationally challenging to predict or factorize. The Euler’s totient function ϕ(n) is calculated for each selected prime product. This function helps to ensure the generated public and private keys are coprime to the totient values. Additionally, it is a necessary condition for their integrity and correctness. The public key components are PK, FK, RN and private key components are QK, GK, RN. This public key can be used for encryption in secure financial data. The private key is derived from calculating the modular inverse of the public key relative to the totient of the prime products. This ensures that the private key can be used to decrypt data that was encrypted using the public key. Throughout the key generation process the algorithm ensures the keys (PK, FK, QK, GK) are all coprime with respect to Euler’s totient values of the prime-based numbers that involved. The integrity of the keys is maintained through regular Greatest Common Divisor (GCD) checks, ensuring that the keys function correctly for secure communication and data protection. The Python programming language is used to implement suggested key generation methodology. The following are some programming code outlines of the random prime number and distinct key generation process.

```
import random
def generate():
seed = random.randint(100,100)
prime = []
for i in range(10,seed):
if(check(i)):
prime.append(i)
return(random.choice(prime))
def genotps(mu):
otps = []
i = 1
while(i<=mu):
sec = generate()
if (sec not in otps):
otps.append(sec)
i+=1
return(otps)
```

3.1. Encryption

Encryption is the process of converting plaintext messages into ciphertext to ensure data confidentiality. In this context, the suggested approach introduces a hybrid encryption algorithm that combines the strengths of the Modified IDEA (M-IDEA) and the Double Secure RSA (DS-RSA) algorithm. The aim of this hybrid approach is to enhance the security and robustness of the encryption process.

3.1.1. Modified IDEA (M-IDEA) Encryption Algorithm

The encryption algorithm operates on a 256-bit plaintext message with a corresponding 512-bit key. The message is divided into four 64-bit sub-blocks, denoted as P1, P2, P3 and P4. The 512-bit key is divided into eight subkeys, each consisting of 64 bits. These sub-keys are denoted as MSK1, MSK2, MSK3, MSK4, MSK5, MSK6, MSK7, MSK8. The algorithm consists of eight identical rounds and a final transformation that is half a round long, employing algebraic and logical operations on 64-bit blocks in each round. In each round, a series of 14 steps involving addition modulo (264), multiplication modulo (264 + 1), and bitwise XOR are executed. The encryption technique using M-IDEA is present in Algorithm 2.

```
Require: P 1, P 2, P 3, and P 4
Ensure: Sub keys MSK1, MSK2, MSK3, MSK4, MSK5, MSK6, MSK7, and MSK8
PSK1 = (P 1 × MSK1) % (264 + 1). // Multiplication modulo operation
PSK2 = (P 2 + MSK2) % 264
PSK3 = (P 3 + MSK3) % 264 // Addition modulo operation
PSK4 = (P 4 × MSK4) % (264 + 1).
PSK5 = (PSK1 ∧ PSK3) // Bit wise XOR operation
```

```
6: $PSK6 = (PSK2 \wedge PSK4)$
7: $PSK7 = (PSK5 \times MSK5) \% (264 + 1)$
8: $PSK8 = (PSK6 + PSK7) \% 264$
9: $PSK9 = (PSK8 \times MSK6) \% (264 + 1)$
10: $PSK10 = (PSK7 + PSK9) \% 264$
11: $PSK11 = (PSK1 \wedge PSK9)$
12: $PSK12 = (PSK3 \wedge PSK9)$
13: $PSK13 = (PSK2 \wedge PSK10)$
14: $PSK14 = (PSK4 \wedge PSK10)$
```

Algorithm 2 M-IDEA Encryption

In every round, excluding the concluding transformation, a swap takes place. The input for the subsequent round is determined by the outcomes of Step 11, Step 13, Step 12, and Step 14, denoted as P1, P2, P3, P4, respectively.

This sequential process repeats for eight rounds, followed by the execution of a ninth "half round". In the final transformation:

1. $PSK1 = (P1 \times MSK1) \% (264 + 1)$. // Multiplication modulo operation

2. $PSK2 = (P2 + MSK2) \% 264$

3. $PSK3= (P\,3 + MSK3) \% 264$ // Addition modulo operation

4. $PSK4= (P\,4 \times MSK4) \% (264 + 1)$.

The resulting blocks are concatenated to produce the ciphertext message (C1). Further enhancing security, a new ciphertext message (C2) is regenerated by re-encrypting the initial ciphertext message (C1) using the DS-RSA algorithm. The complexity of the operations ensures that even if an attacker has knowledge of the algorithm, it is computationally infeasible to reverse the transformations and retrieve the original plaintext without knowing the decryption keys.

3.1.2. Double Secure RSA (DS-RSA) Encryption Algorithm

In the process of double encryption employing the DS-RSA algorithm, the ciphertext message C1 undergoes encryption through the application of the equations:

$$CK2 = (C1)PK \%RN$$

$$C2 = (CK2)FK \%RN$$

In this context, C1 signifies the ciphertext message generated by the M-IDEA algorithm. The associated public key is denoted as (PK, FK, RN). The CK2 signifies the first part of the ciphertext message using the DS-RSA algorithm. C2 symbolizes a fortified ciphertext message, thereby enhancing the security of the original communication.

```
Require: PK, FK, C1, RN
Ensure: $CK2 = (C1PK) \%RN$ , $C2 = (CK2FK) \%RN$
1: Take the ciphertext message C1
2: Filep1= ConvertASCII (C1)//Convert character into ASCII values in binary
3: Take block size denoted as PS.
4: Filep12 = Decimal(Filep1) // According to block size PS.
5: Utilize a public key (PK, FK, RN ) for encryption.
6: PSZ = ceil(log2(RN )) // number of bits to represent an intermediate changed
7: $CK2 = (Filep1PK)\%RN$ //sequentially reading decimal values
8: $C2 = (CK2)FK\%RN$
9: Filep2 = Binary(C2) // Convert the intermediate output into a binary
10: length = Count(Filep2)
11: if (length%8 == 0) then
12:     No need to add dummy bits at the end of Filep2
13: else
14:     Add dummy bits at the end of Filep2
15: end if
16: while $SC8 \leq length$ do
17:     SCP8 = Decimal(SC8)
18:     Symbol = ConvertSymbol(SCP8)
19:     Filep3 = Filep3 + Symbol
20: end while
21: Exit
```

Algorithm 3 DS-RSA Encryption

Algorithm 3 introduces encryption technique using DS-RSA methodology. The message C1 is the initial input ciphertext that will undergo the encryption process. Convert each character in the message C1 to its corresponding ASCII value in binary. Define the block size PS determines amount of data is processed at once. It is based on system requirements or the size of the input data. Convert the binary ASCII values of C1 into decimal form based on the defined block size PS. The public key component (PK) will be used to encrypt the data. This involves using RSA modular arithmetic. Calculate the bit size PSZ, which represents the number of bits needed to express the modulus RN in binary. Perform modular exponentiation on the Filep1 (the decimal-encoded message) with the public key PK, and the modulus RN. This gives the intermediate value CK2. Now apply the public key component FK to the intermediate result CK2 using modular arithmetic again. The result C2 is the final encrypted message. Determine the length of the binary output Filep2 to check if it needs padding. Add extra bits (usually 0's) at the end of the binary string to make the length a multiple of 8. Append the decoded symbols to the final output (Filep3), reconstructing the encrypted message in its original format. A ciphertext message (Filep3) as C2 has been sent to the recipient. The user receives the original message or file along with the key pair over a secure channel after it has been encrypted. The Python programming code of the encryption process is given below.

```
def Encrypyion(self):
try:
intermediate_binary_file = open(self.inter_file_name,"rb")
except Exception as e:
print(e+" error"+self.inter_file_name)
t_data_holder=[]
bunch_size = 4
for i in intermediate_binary_file.read():
if(len(t_data_holder)==bunch_size):
ttt=""
for i_temp in t_data_holder:
ttt+=chr(i_temp)
t__ =int(ttt,2)
self.Input_data_in_binary_coded_ascii_format.append(t__)
t_data_holder=[i]
else:
t_data_holder.append(i)
```

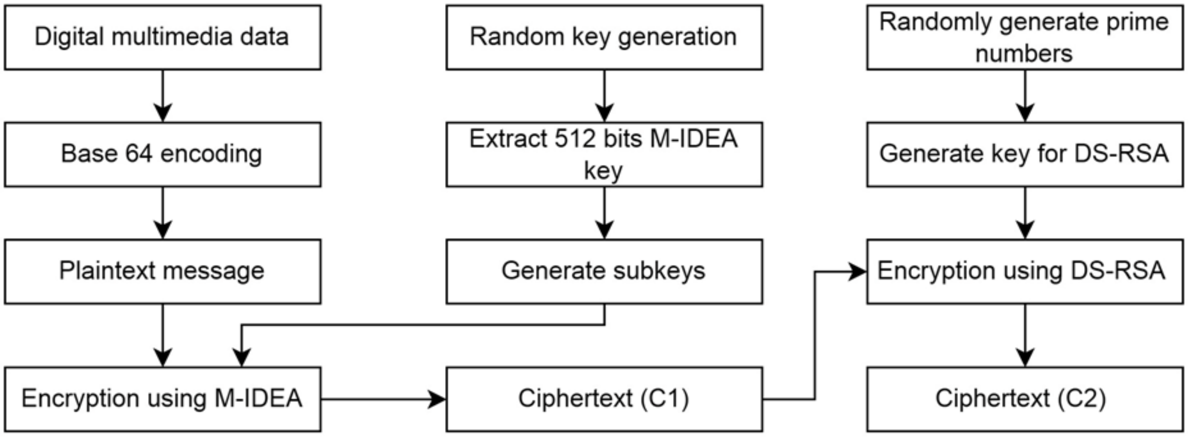

Figure 1 Workflow Diagram of Hybrid Encryption Technique

Figure 1 illustrates the workflow diagram of the proposed hybrid encryption technique. The process flow diagram of the proposed hybrid encryption technique is represented in Figure 1. The initial step involves converting the data into a base64 encoded text format. The base64-encoded version undergoes the first encryption using the proposed M-IDEA algorithm with a 512-bit key size. The key generation methodology is used to generate 512-bit keys for the encryption and decryption process. This key is used in the proposed M-IDEA algorithm for the first level of encryption. During the key generation process, it performed bitwise XOR and bit left circular shifts operations which adds an extra layer of security in key generation. After that, it is used the proposed DS-RSA algorithm for encrypted data. It generates private and public keys for the encryption and decryption data on both ends. The proposed DS-RSA algorithm uses the ‘N’ prime number, the Euler totient function, and arithmetic operations such as the greatest common divisor for key generation. It randomly generates ‘N’ larger prime numbers. Plaintext messages are first encrypted using the proposed M-IDEA algorithm. This resulting cipher text message is represented as C1. Subsequently, it undergoes re-encrypting using the DS-RSA algorithm to generate a new ciphertext as C2. This framework provides a robust security solution by leveraging both M-IDEA for fast, efficient encryption of data and DS-RSA for safeguarding the data during transmission and protecting it against various cyber threats.

3.2. Decryption

The decryption process is characterized by the transformation of ciphertext messages into plaintext. Initially, the ciphertext message C2 undergoes conversion into a plaintext message

PC1 using the DS-RSA algorithm. Subsequently, the resulting plaintext PCK1 is further transformed into the original plaintext using the M-IDEA algorithm. This hybrid approach is strategically designed to elevate the overall security and resilience of the decryption process during online transactions in the financial sector.

3.2.1. Double Secure RSA (DS-RSA) Decryption Algorithm

The decryption of the ciphertext message C2 is accomplished through the utilization of the DS-RSA decryption algorithm, employing the equation

PCK1 = (C2)GK%RN

PC1 = (PCK1)QK%RN

In this context, PCK1 denotes the initial plaintext message derived through the DS- RSA algorithm, utilizing the private key components GK, QK, RN which are formed through the Algorithm 1.

```
Require: Private key components GK, QK, RN
Ensure: PCK1 = (C2)GK%RN and PC1 = (PCK1)QK%RN
Read each character of the ciphertext message C2 from Filep3
Filep4 = Convertbinary(C2), // Character into its corresponding 8-bit binary
Read Block size as PS. //Block size is the same in encryption and decryption.
PSZ = ceil(log2(RN )
Use private key components GK, QK, RN
PCK1 = (Filep4)GK%RN
PC1 = (PCK1)QK%RN
Filep5 = Binary(PC1) // Convert the intermediate output into a binary
length = Count(Filep5)
while DC8 ≤ length do
    DCP 8 = Decimal(DC8)
    Symbol = ConvertSymbol(DCP 8)
    Filep6 = Filep6 + Symbol
end while
Exit
```

Algorithm 4 Decryption Using DS-RS

Algorithm 4 introduces decryption using proposed DS-RSA methodology. The encrypted message C2 is read from the file Filep3. Convert each character of the ciphertext C2 into its corresponding 8-bit binary representation. The same block size PS is used in decryption and encryption process. This block size determines how much data is processed at once during decryption. Calculate the bit size PSZ, which represents the number of bits needed to express the modulus RN in binary. This is important for handling intermediate values in modular arithmetic. The private key components GK and QK, along with the modulus RN, are used to decrypt the message. Apply the first private key component GK to the binary-encoded ciphertext Filep4 using modular exponentiation (Filep4GK) % RN. This produces the intermediate decrypted value PCK1. Apply the second private key component QK to the intermediate value PCK1 using modular exponentiation (PCK1QK) % RN. This gives the final decrypted message PC1. Convert the decrypted value PC1 into its binary form to proceed with further processing. Calculate the length of the binary output Filep5. This helps determine whether padding is needed or if the data is properly formatted. Process the binary data in 8-bit chunks (DC8) until the entire message is processed. Convert each 8-bit chunk (DC8) into its corresponding decimal value. Map each decimal value back to its first level of data format. Append the decoded symbols to the final output Filep6, reconstructing the decrypted message step by step. Continue processing until all chunks of the binary data have been decoded and the message has been fully decrypted. The following is a snippet of pseudocode illustrating the decryption technique.

```
def decryption (self):
self.Input_data_in_binary_decoded_ascii_format=[]
for i in self.Encript_Message_ascii_format_temp:
ttt=i
m = (ttt**d)% n
self.Input_data_in_binary_decoded_ascii_format.append(m)
for i in temp_string:
if(len(t_data)==8):
ttt=chr(int(t_data,2))
self.the_read_message.append(ttt)
t_data=i
else:
t_data+=i
```

3.2.2. Modified IDEA (M-IDEA) Decryption Algorithm

The process of converting encrypted data back into its original readable form is referred to as decryption. The plaintext message is recovered at this point once the ciphertext message is received and processed using the suggested M-IDEA algorithm. The subsequent paragraph delineates the specific steps involved in the decryption process.

The initial output serves as the primary ciphertext Filep6 as C1, which is subsequently subjected to re-decryption using the M-IDEA algorithm, ultimately yielding the original plaintext message.

The implementation encompasses all eight and a half rounds of the M-IDEA algorithm, covering essential functionalities such as addition modulo, multiplication modulo, and bitwise XOR.

The decryption process mirrors the encryption process in its entirety.

The sole distinction lies in the sub-key rules, where the encryption process involves reversing the order of the sub-key, and the sub-key itself is inverted.

During the output transformation step in the encryption process, the sub-key is invoked and repurposed as a sub-key in the first round of the decryption process.

Upon the completion of all rounds, the original plaintext message is successfully regenerated.

The decryption process is meticulously crafted to undo the encryption effects and restore the original plaintext. It leverages inverse operations corresponding to those employed in the encryption process to accomplish this reversal. The mathematical operations incorporated are purposefully structured to introduce security, confusion, and diffusion throughout the key generation, encryption, and decryption methodology.

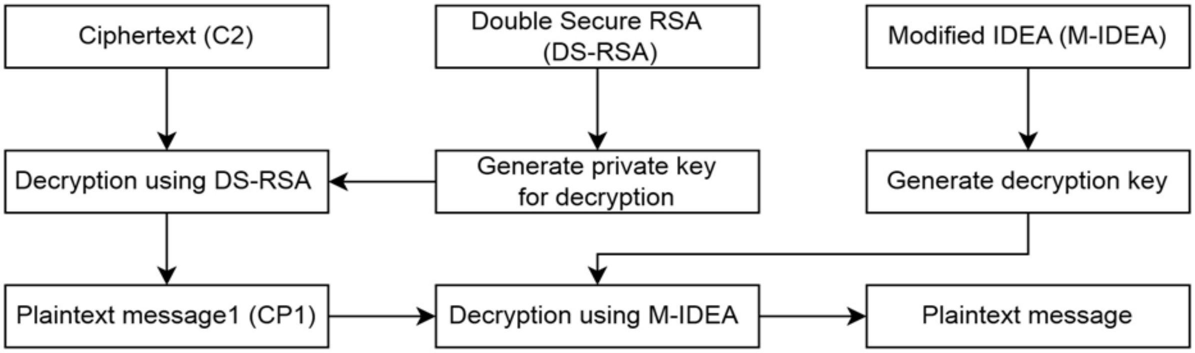

Figure 2 Workflow Diagram of Hybrid Decryption Technique

Figure 2 illustrates the decryption process in a hybrid cryptographic framework combining Double Secure RSA (DS-RSA) and Modified IDEA (M-IDEA) methodology. The decryption process intricately reverses the encryption process involving a slightly complex methodology. The decryption process begins with ciphertext (C2), which is the final encrypted output from the encryption process that used both DS-RSA and M-IDEA. Initially, for the decryption process the keys regenerate by using proposed M-IDEA and DS-RSA algorithm. After that, decrypting ciphertext message C2 back to the first plaintext message CP1 using the private key of the DS-RSA algorithm. Furthermore, the CP1 message undergoes secondary decryption using the M-IDEA algorithm.

The output of the M-IDEA decryption is the final Plaintext message, representing the original data before encryption. This completes the decryption process, successfully retrieving the original data. This dual-layer decryption ensures enhanced data security, where DS-RSA secures the outer encryption layer and M-IDEA secures the inner layer, effectively reconstructing the original plaintext message from the doubly encrypted data.

4. RESULTS AND DISCUSSION

This section demonstrates the outcomes achieved through the implementation of the proposed algorithms. The suggested methodology delivers an exceptionally high level of data protection using the hybrid cryptography approach combined with M-IDEA with 512-bit key size and DS-RSA techniques. It ensures a peak level of security for key generation, encryption, and decryption techniques. The proposed method has been applied to various large-sized files, including text, image, audio, PDF, and executable files of different sizes. It demonstrates the encryption time using the suggested algorithms across these diverse file types. A key feature of this approach is that the encryption time varies for each file type, independent of the key size and the number of prime numbers used. The resulting ciphertext has undergone two levels of mixed encryption, incorporating M-IDEA with 512-bit key size and the DS-RSA algorithm. This hybrid encryption technique significantly improves security compared to individual applications of a single model.

4.1. Implementation and Simulation Environment

The proposed hybrid cryptography methodology was implemented in Python 3.11. The simulation environment consists of a 64-bit Microsoft Windows 8.1 operating system. Key hardware and software specifications are summarized in Table 2.

Table 2 Configuration Details of the Simulation Environment

Simulation parameters	Configuration
Processor	Intel ® Core™ i3- 2370 M CPU @ 2.40 GHz
Operating system	Microsoft Windows 8.1
System type	64 – bit Operating system, X64-based processor
Hard disk	1 TB
RAM	8.00 Gigabytes DDR4
Programming platform	Python 3.11

This configuration provides a stable platform for testing the proposed hybrid cryptography methodology. It ensures that the system meets the computational requirements for the encryption and decryption processes involved in the research.

A table that details the key generation times for the hybrid methodology with respect to various file sizes and number of prime numbers is shown in Table 3. The first column of the table denotes the data file size in megabytes which varies from 0.29 to 3.60 MB. The second column of the same table represents the number of prime numbers between 2 to 10. The third column from the left in the same table shows how long it takes to generate a key with the proposed DS-RSA technique. This time frame has exhibited variability, ranging from 0.06432 to 0.20635 seconds. The fourth column from the left denoted key generation time for M-IDEA with the 512-bit key size which varies between 0.54224 to 3.62841 seconds. The key generation time of the hybrid methodology combined with the M-IDEA and DS-RSA techniques is presented in the last column. The results of the key generation time unequivocally reveal that the duration for the hybrid methodology exhibits variation within the range of 0.60656 to 3.83477 seconds.

Table 4 demonstrates the encryption times for various kinds of file sizes and types using the suggested hybrid approach. The names and types of data files used for the process of encryption are listed in the first column on the left side of Table 4, and the size of each data file in megabytes is shown in the second column of the corresponding table. The time required to encrypt data using the M-IDEA approach with a 512-bit key size is shown in the third column from the left of the same table. In the experiment, to decrypt the data it takes the time ranged from 0.24628 to 8.01745 seconds. The table also represents encryption times for the DS-RSA technique which is shown in the fourth column and varies from 0.38074 to 11.67602 seconds. The last column denotes the hybrid encryption time of the proposed methodology. It has been demonstrated that the time taken to encryption per byte varies from 0.62702 to 19.69348 seconds. It clearly shows the results of the hybrid cryptography approach is a secure and efficient encryption process for various file types and sizes.

Table 3 Key Generation Time of the Proposed Hybrid Methodology

Data file size (MB)	Number of prime numbers	Key generation time in DS-RSA (second)	Key generation time in M-IDEA (second)	Key generation time in hybrid methodology (second)
0.29	2	0.06432	0.54224	0.60656
0.57	3	0.07723	0.90143	0.97866
0.86	4	0.15765	1.06956	1.22722
1.14	5	0.17024	1.96941	2.13965
1.72	6	0.17869	2.00744	2.18613
1.99	7	0.18105	2.28575	2.46680
2.80	8	0.19151	2.82301	3.01452
3.20	9	0.19772	3.12852	3.32625
3.60	10	0.20635	3.62841	3.83477

Table 4 Encryption Time of the Proposed Hybrid Methodology

Data file name and type	Data file size (MB)	Encryption time for M-IDEA 512-bit key size (second)	Encryption time for DS-RSA (second)	Hybrid encryption time (second)
signature.png	0.12	0.24628	0.38074	0.62702
bank.pdf	0.29	0.62536	0.92012	1.54548
account.docx	0.32	0.68416	1.01531	1.69947
login.exe	0.57	1.22249	1.80851	3.03100
aadhar.jpg	0.70	1.50132	2.22098	3.72230
shoping.png	0.86	1.86052	2.72864	4.58915
atm.pdf	0.98	2.10925	3.10938	5.21862
goldloan.doc	1.14	2.45549	3.60434	6.05983
credit.mpeg	1.72	3.72017	5.45727	9.17744
debit.docx	1.99	4.33012	6.31394	10.64405
demo.mpeg	2.80	6.09363	8.88393	14.97756
registration.exe	3.68	8.01745	11.67602	19.69348

Table 5 Decryption Time of the Proposed Hybrid Methodology

Data file name and type	Data file size (MB)	Decryption time for M-IDEA 512-bit key size (second)	Decryption time for DS- RSA (second)	Hybrid decryption time (second)
signature.png	0.12	1.41065	0.96773	2.37838
bank.pdf	0.29	3.68068	2.33869	6.01937
account.docx	0.32	4.03486	2.58062	6.61548
login.exe	0.57	7.26719	4.59673	11.86392
aadhar.jpg	0.70	8.93608	5.64511	14.58118
shoping.png	0.86	11.09968	6.93542	18.03509
atm.pdf	0.98	12.64422	7.90315	20.54737
goldloan.docx	1.14	14.64331	9.16120	23.80451
credit.mpeg	1.72	22.21083	13.87084	36.08167
debit.docx	1.99	25.80378	16.04823	41.85201
demo.mpeg	2.80	36.30403	22.58043	58.88446
registration.exe	3.68	47.75435	29.67714	77.43148

Table 5 showcases decryption times for various file types and sizes by using the hybrid cryptography approach. It has been demonstrated that the time taken to decrypt data files varies from 1.41065 to 47.75435 seconds using the M-IDEA technique with 512-bit for data file sizes 0.12 to 3.68 MB. Decryption time for DS-RSA techniques varies between 0.96773 to 29.67714 seconds. The time taken to decrypt data files for hybrid techniques varies from 2.37838 to 77.43148 seconds. The results illustrate that as the complexity and size of the data file increases, so does the decryption time across all methods, with the hybrid method consistently taking the longest. This demonstrates the trade-off between enhanced security through a hybrid approach and the increased time required for decryption.

The proposed model achieves better results through an innovative and efficient hybrid cryptography approach for key generation, encryption, and decryption techniques. The methodology strikes an optimal balance between high security and performance by integrating M-IDEA and DS-RSA. The use of a 512-bit key with variable block sizes enhances flexibility and scalability. It allows the model to adapt to varying levels of security and data sizes. Selecting randomly distinct prime numbers in the key generation process ensures both randomness and unpredictability against factorization attacks. The hybrid encryption methodology that combines M-IDEA with DS-RSA offers enhanced security due to its multiple layers of defense. The proposed model achieves an encryption time of 1.54548 seconds for 0.29 MB of data, showcasing an excellent balance between speed and security, particularly for larger datasets. The extended decryption time due to double decryption process introduces an additional layer of complexity for potential attackers. The combination of different mathematical operations at multiple stages in the encryption and decryption process adds significant layers of complexity. The inclusion of both algorithms allows for a highly secure and robust system that is resilient to a wide range of attacks, including man-in-the-middle and data disclosure attacks. This hybrid approach ensures that both performance and security are optimized, with the added flexibility to handle sensitive data securely in environments like secure e-banking and financial transactions.

5. PERFORMANCE ANALYSIS AND COMPARISON

The purpose of this section is to compare and analyze the proposed methodology for key generation, encryption, and decryption time. A comparison of time analysis is presented between the RSA [3], modified RSA [40], and proposed hybrid approach in the subsequent tables. The tables and graphs of this section provide insights into their respective performance across a variety of data types and sizes. Further evaluation is demonstrated through tables and graphs, providing a comprehensive overview of algorithms' performance metrics, including key generation time, encryption time, and decryption time.

A graphical representation of Table 3 is shown in Figure 3 with a continuous solid line, dotted line, and dashed line. The graph illustrates clearly that as the size of the data file and the number of prime numbers increase then the key generation time with the hybrid methodology consistently increases. The key generation technique generates prime numbers randomly that are also affected in the graph. The hybrid methodology analyzes that when the number of prime numbers increases, key generation time can sometimes increase and sometimes decrease. The reason for this is that key generation time is dependent on prime number values rather than number of prime numbers.

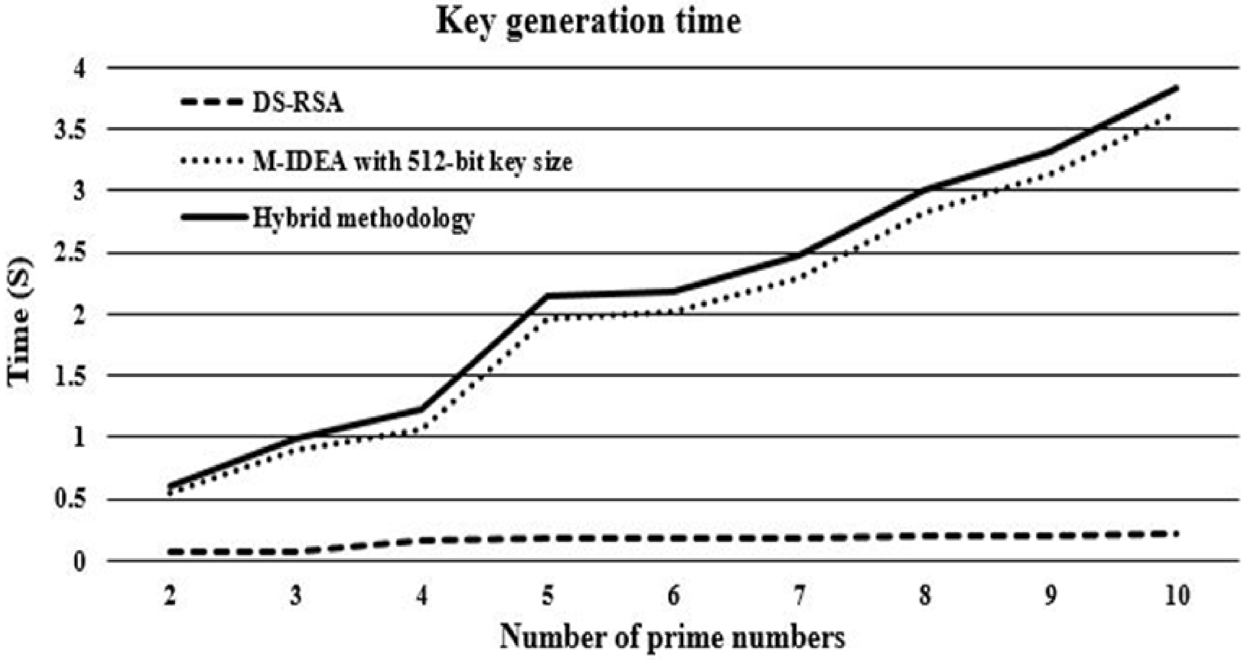

Figure 3 Key Generation Time of M-IDEA, DS-RSA, and Hybrid Methodology

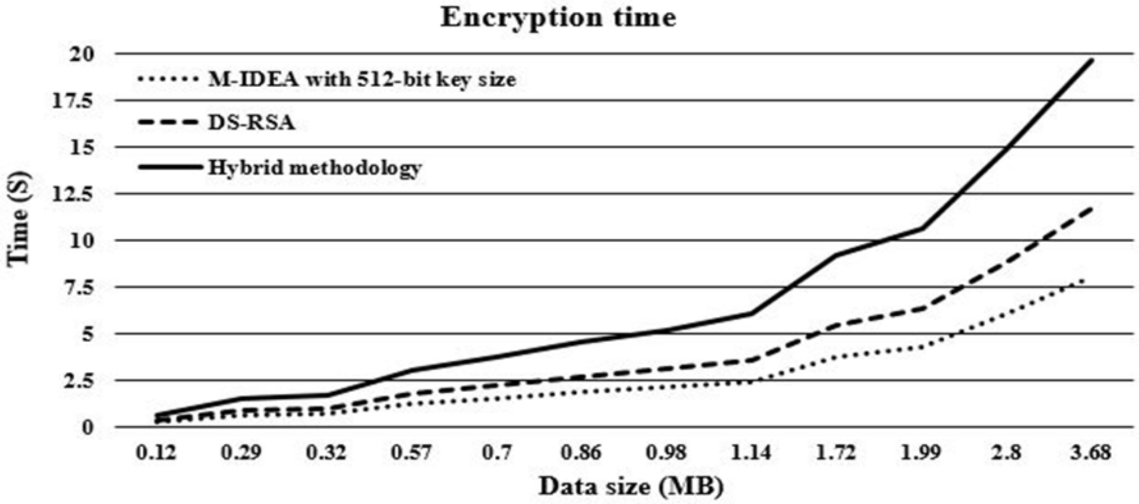

Figure 4 File Size Versus Encryption Time of M-IDEA, DS-RSA, and Hybrid Methodology

The graphical representation of Table 4 is displayed in Figure 4 through both a continuous solid line, a dashed line, and a dotted line. The solid line is used to denote encryption time using hybrid methodology. The dashed line signifies encryption time using M-IDEA with 512 bits key size and the dashed line is used to represent encryption time using the DS-RSA algorithm. Notably, the graph indicates a striking similarity between the three lines. The encryption time gradually increases when the file size is increased. It has been observed from the graph DS-RSA algorithm takes higher encryption time than the M-IDEA algorithm. It becomes apparent that the prime numbers are generated randomly which is affected by the graph. This exponential escalation is evident in both scenarios, whether involving a randomly selected private key or a specific value of the private key. Nevertheless, the preference for random private key values contributes to bolstering the algorithm’s security. The examination of time complexity serves to underscore the practicality and viability of implementing our proposed method. The hybrid encryption method significantly increases encryption time, suggesting a trade-off between enhanced security and performance. The increased time reflects the additional complexity and computational demands of using two encryption methods together. This trade-off is crucial for applications where encryption speed is as important as security, indicating that while the hybrid method may offer superior security, it might not be suitable for time-sensitive applications.

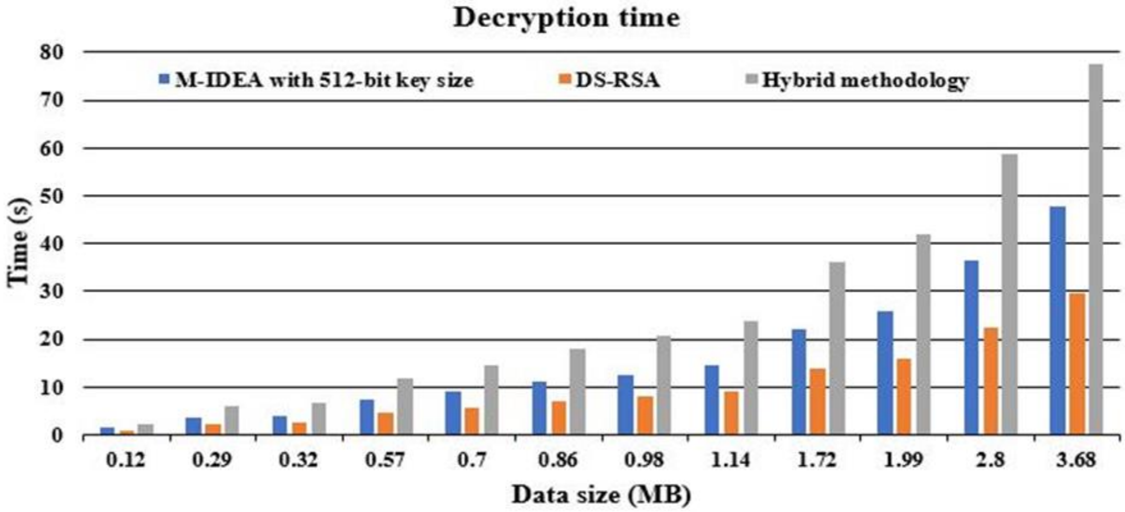

Figure 5 File Size Versus Decryption Time of M-IDEA, DS-RSA, and Hybrid Methodology

The findings presented in Figure 5 derived from the data in Table 5 are visually represented through a bar diagram. The decryption times associated with suggested cryptographic algorithms are distinguished by color. The blue bars correspond to the decryption time using M-IDEA with a 512-bit key size, while the orange bars represent the decryption time employing the DS-RSA algorithm. The grey bars signify the decryption time when employing a hybrid methodology. Notably, the graphical representation reveals that M-IDEA with a 512-bit key size exhibits a longer decryption time compared to the DS-RSA algorithm. Consequently, proposed analysis leads to the conclusion that the proposed hybrid methodology demonstrates a higher decryption time when compared to both M-IDEA and DS-RSA algorithms. Importantly, the prolonged decryption time contributes to enhanced algorithmic security, rendering the system more resistant to unauthorized access and thereby increasing its resilience against potential attackers. The results suggest that while the hybrid method may offer superior security, it also demands more processing time, which is an important consideration for real-time or performance-critical applications. The research indicates the effectiveness of the hybrid method in securing data but also highlights the need for optimizing decryption time to balance security with efficiency.

The encryption time and decryption time of proposed hybrid methodology are shown in Figure 6. In this representation, the blue bars correspond to the encryption time achieved and the orange bars illustrate the decryption time associated with hybrid methodology. Consequently, it can be inferred that the proposed hybrid methodology incurs a higher decryption time compared to the encryption time using the hybrid methodology. This prolonged decryption time contributes to enhanced algorithmic security, making the system significantly more resilient against potential attacks.

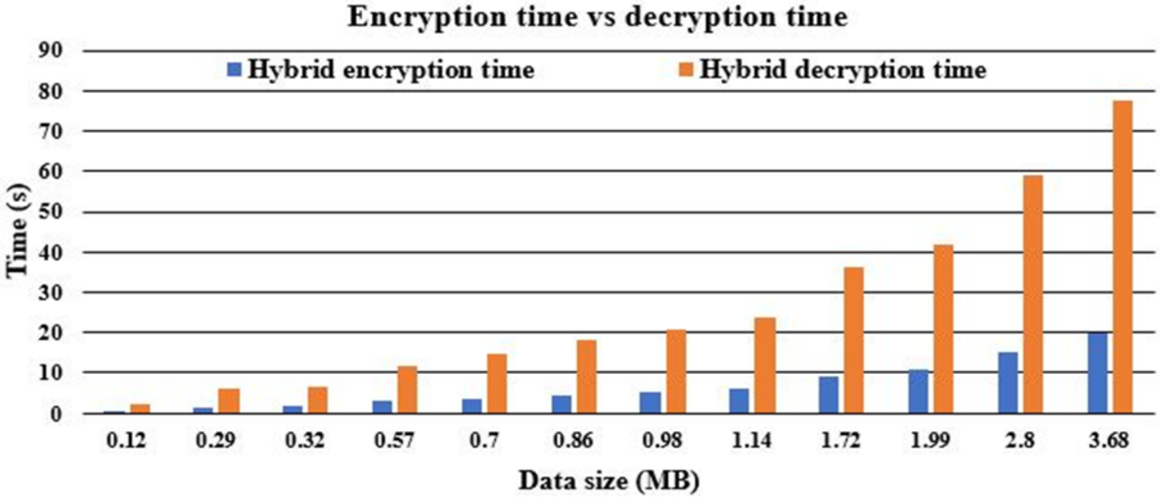

Figure 6 Encryption Time Versus Decryption Time of Proposed Hybrid Methodology

Table 6 Comparison of Key Generation Time in RSA, Modified RSA, and Hybrid Methodology

Data file size (MB)	RSA (second)[3]	Modified RSA (second)[40]	Hybrid methodology (second)
0.29	0.0013	0.00069	0.60656105
0.57	0.0013	0.00066	0.97865815
0.86	0.0012	0.00042	1.22721650
1.14	0.0014	0.00043	2.13965130
1.72	0.0013	0.00043	2.18612865
1.99	0.0013	0.00044	2.46679870
2.80	0.0015	0.00044	3.01452335
3.20	0.0015	0.00046	3.32624641
3.60	0.0016	0.00074	3.83476540

Table 6 presents a comparison of key generation times across three cryptographic methodologies such as RSA [3], Modified RSA [40], and Hybrid methodology across various data file sizes. The first column includes a range of data file sizes from 0.29 MB to 3.60 MB. The subsequent column details the time required for key generation through the conventional RSA algorithm, showcasing a variance from 0.0013 to 0.0016 seconds. Following that, the third column illustrates key generation times utilizing the modified RSA algorithm. The fourth column delineates the key generation duration using the suggested hybrid methodology, encompassing a range of 0.60656105 to 3.83476540 seconds. The hybrid methodology, on the other hand, shows significantly higher key generation times compared to both RSA and Modified RSA. The key generation time increases more noticeably with the size of the data file, indicating that this method’s complexity scales with file size. This trade-off might be acceptable or even desirable in high-security environments where the integrity and confidentiality of the data are paramount, and the additional time required for key generation is a worthwhile investment.

Table 7 showcases the encryption times associated with the traditional RSA methodology, modified RSA techniques, and the newly proposed hybrid methodology, considering diverse file types and sizes. Encryption time is a crucial metric that impacts the efficiency and performance of cryptographic processes. The initial column denotes a range of data file sizes from 0.29 MB to 2.80 MB, allowing for an examination of how encryption time scales with file size for each method. The second column highlights the time required for encryption using the traditional RSA algorithm, demonstrating variability ranging from 0.00286 to 0.02425 seconds. Subsequently, the third column illustrates the modified RSA algorithm’s encryption times from 0.00511 to 0.04456 seconds. The fourth column delineates the encryption durations using the suggested hybrid methodology, ranging from 1.54548 to 14.97756 seconds.

Table 7 Comparison of Encryption Time in RSA, Modified RSA, and Hybrid Methodology

Data file size (MB)	RSA (second) [3]	Modified RSA (second) [40]	Hybrid methodology (second)
0.29	0.00286	0.00511	1.54548
0.57	0.00522	0.00713	3.03100
0.86	0.00727	0.00919	4.58915
1.14	0.00950	0.01840	6.05983
1.72	0.01403	0.02947	9.17744
1.99	0.01612	0.03053	10.64405
2.80	0.02425	0.04456	14.97756

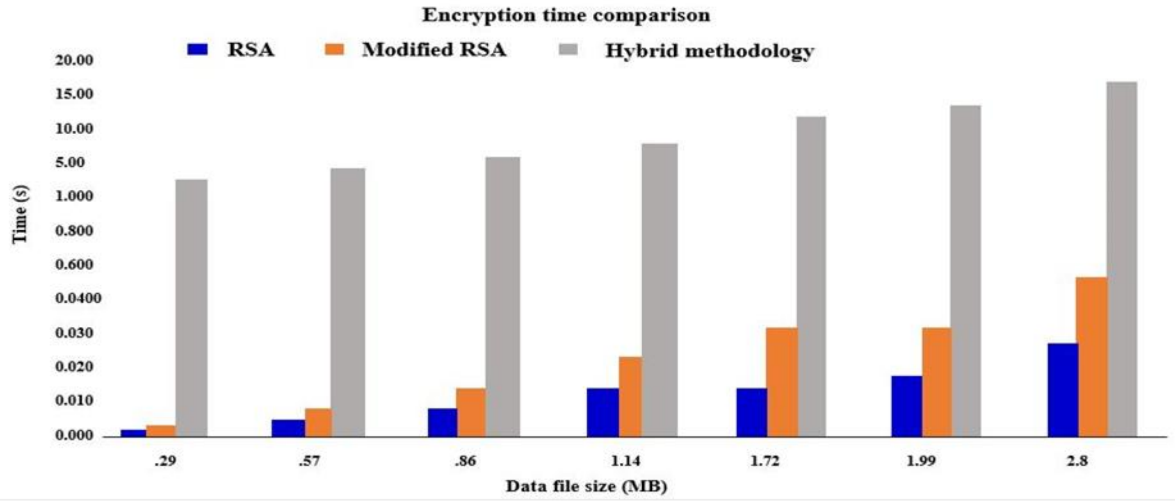

Figure 7 Comparison of Encryption Time Between RSA, Modified RSA, and Hybrid Methodology

Table 7 reveals that the proposed hybrid methodology incurs a higher encryption time compared to the traditional and modified RSA algorithms. This observed increase in encryption time which can be attributed to the double encryption process inherent in the proposed methodology. Moreover, RSA and Modified RSA, with their lower encryption times, are well-suited for environments where speed and efficiency are critical, and the security provided by these methods meets the application's requirements. The Hybrid methodology, on the other hand, is more appropriate for high-security applications where the added encryption time is acceptable in exchange for potentially higher security levels.

Figure 7 presents a comprehensive analysis of encryption times for the hybrid methodology, integrating M-IDEA and DS-RSA algorithms, alongside traditional RSA and modified RSA algorithms. The evaluation of time efficiency emphasizes the practical viability of implementing the proposed method in real-world applications. Noteworthy is the dual-layered encryption applied to the resulting ciphertext, involving both M-IDEA and the DS-RSA algorithm. This hybrid encryption approach notably enhances the security of online E-banking transactions compared to the singular use of a standalone model, thereby reinforcing the robustness of the suggested method.

Table 8 presents the decryption times associated with the traditional RSA methodology, modified RSA techniques, and the newly proposed hybrid methodology, considering diverse file types and sizes. This comparison offers valuable insights into the performance characteristics of each methodology, particularly in the context of decryption, which is a critical operation in the realm of data security and accessibility. The leftmost column indicates the size of the data files in megabytes. The second column highlights the duration required for decryption using the traditional RSA algorithm, demonstrating variability ranging from 0.01624 to 0.05876 seconds. Moving to the third column, it showcases the decryption times using the modified RSA algorithm, fluctuating from 0.08353 to 0.77434 seconds. The fourth column outlines the decryption durations using the suggested hybrid methodology. The Hybrid methodology exhibits significantly higher decryption times across all data sizes, from 6.01937 seconds for the smallest file to 58.88446 seconds for the largest file. This stark increase indicates a highly complex decryption process, likely integrating multiple security measures or algorithms to achieve a superior level of data protection.

Notably, the table reveals that the proposed hybrid methodology incurs a higher decryption time compared to the traditional and modified RSA algorithms. This heightened decryption time is attributed to the double decryption process inherent in the proposed methodology. Importantly, this increase in decryption time contributes to the algorithm's enhanced hardness, thereby bolstering the security robustness of the proposed methodology. The elevated decryption time poses a significant challenge for potential attackers, making it more difficult to breach the system and ensuring a higher level of security. The Hybrid methodology provides the highest level of security, suitable for applications where data protection is the highest priority, and decryption speed is a secondary concern.

Table 8 Comparison of Decryption Time in RSA, Modified RSA, and Hybrid Methodology

Data file size (MB)	RSA (second) [3]	Modified RSA (second) [40]	Hybrid methodology (second)
0.29	0.01624	0.08353	6.01937
0.57	0.02127	0.15745	11.86392
0.86	0.01918	0.23660	18.03509
1.14	0.02460	0.31220	23.80451
1.72	0.03604	0.47068	36.08167
1.99	0.04474	0.47147	41.85201
2.80	0.05876	0.77434	58.88446

Table 9 Comparison Among Proposed Hybrid Methodology Using M-IDEA and DS-RSA and Existing Works

Evaluation Factors	MIDEA with the 1024-Bit Key Size [41]	Hybrid Methodology Using RSA and SSK [28]	Proposed Hybrid Methodology Using M-IDEA and DS-RSA
Key size (bit)	1024	4 - 512	512
Block size (bit)	Two 512-bit blocks, each divided into 8 sub-blocks of 64-bit	Variable	Variable
Key generation time (second)	47.61170	0.007	0.60656105
Encryption time (second)	6.00607 for 0.29 MB data	0.0065 for 300 bit message length	1.54548 for 0.29 MB data
Decryption time (second)	114.73961	0.0055	6.01937
File types supported	Text, image, audio, video, PDF, Word.	Any type of data	Any type of data
Security level	High	4.5	Very high
Efficiency	High	Better	Very high
Protection against Man-in-the- Middle attack	-	-	Prevention possible
Protection against data disclosure breaches attack	-	-	Prevention possible

Table 9 introduces a comparative analysis of the proposed hybrid methodology using M-IDEA and DS-RSA against MIDEA with the 1024-bit key size [41] and hybrid methodology using RSA and SSK [28]. According to comparative results MIDEA employs a large 1024-bit key size which contributes to its robust security but may impact efficiency due to higher processing times. The proposed hybrid methodology uses a 512-bit key size which is balancing security and efficiency for moderate-to-high-security applications. MIDEA has significantly longer key generation and decryption times that indicates a trade-off for enhanced security. The proposed hybrid methodology offers moderate key generation and decryption times, improving efficiency while maintaining strong security. The hybrid methodology combines RSA and SSK using smaller key size. It demonstrates the fastest times across key generation, encryption, and decryption but at the cost of reduced security robustness. The MIDEA maintains high security but at a slightly lower efficiency due to longer processing times. The RSA and SSK methodology is relatively efficient but offers moderate security, suitable for less critical applications where speed is a priority. The suggested methodology provides proactive defenses against man-in-the-middle and data disclosure attacks, which are not explicitly addressed by MIDEA or RSA and SSK. The hybrid methodology using M-IDEA and DS-RSA achieves a strong balance between security and efficiency, addressing potential vulnerabilities while maintaining acceptable encryption and decryption times. This makes a highly suitable choice for applications where robust security is essential, alongside the capability to prevent specific types of cyber threats.

6. SECURITY ANALYSIS AND APPLICATION

The security analysis of the hybrid cryptographic algorithm involves examining various aspects of its design, implementation, and operational environment to identify potential vulnerabilities and assess its resilience against attacks. Performing a security analysis of the proposed hybrid methodology using M-IDEA and DS-RSA techniques is crucial to ensuring the robustness and reliability of these cryptographic techniques. The proposed methodology evaluates the longer key size in the DS-RSA and 512-bit in the M-IDEA methodology. Longer key lengths and computation complexity of the suggested methodology in key generation, encryption, and decryption enhance the robust security of the proposed technique.

6.1. Man-in-the-Middle Attack

Man-in-the-middle attack occurs when an attacker intercepts communication between the sender and receiver. The attacker represents himself as one of them. The attacker can silently monitor the communication channel and retrieve, alter, or delete messages with a delay. This enables the attacker to eavesdrop on the communication channel, and potentially retrieve, alter, or delete messages with a delay. To mitigate this, the proposed scheme requires any new user to register and authenticate within the financial application. The suggested hybrid cryptography technique is used throughout the registration process to guarantee safe initial communication, confirming the user's identity and public key pair and preventing any adversary from modifying this data. The public key components are PK and FK in the proposed methodology. The public key is randomly generated based on the conditions GCD(PK, ϕ(xm) × PK12) = 1 and (1 <PK < (ϕ(xm)×PK12)). The value of FK is calculated as GCD(FK, ϕ(xm)) = 1 and (1 < FK < GK < ϕ(xm)). On the other hand, calculate the private key components, QK is calculated using QK = PK−1 % (ϕ(xm)×PK12). Additionally, a value GK is generated such that GCD(GK,ϕ(xm)) = 1 and (1 < GK < ϕ(xm)). Both QK and GK are components of the private key and are generated on the financial server's end. To generate the components of the private key the attacker needs to generate ϕ(xm) which is calculated as ϕ(xm) = ϕ(x1)×ϕ(x2)×ϕ(x3)×ϕ(xn). The ϕ(x1), ϕ(x2), ϕ(x3), ϕ(xn) calculate on ϕ(x1) = (a1−1)×(a2−1), ϕ(x2) = (a3−1)×(a4−1), ϕ(x3) = (a5−1)×(a6−1) and ϕ(xn) = (a7−1)×(an−1). Where a1, a2, …. an are larger prime numbers. Due to the complexity of generating ϕ(xn) through backtracking, it is highly difficult for an attacker to derive QK, GK, and RN to decrypt the encrypted secret content key during transmission. Since the attacker does not know the receiver's private key, decrypting the communication is impossible. Thus, it is infeasible for an intruder to carry out a man-in-the-middle attack.

6.2. Sensitive Data Disclosure Breaches Attack

Sensitive data disclosure breaches represent a serious threat to individuals, organizations, and society at large. These breaches occur when confidential, private, or sensitive information is exposed to unauthorized individuals or entities. The impact of such breaches can be profound, affecting individuals' privacy, the financial sector, businesses, and the government sector. The proposed scheme aims to prevent sensitive data disclosure breaches by encrypting data values using a hybrid cryptography methodology. The financial data values are first encrypted through the M-IDEA algorithm which employs a series of mathematical operations including multiplication modulo, addition modulo, and XOR operations. The multiplication modulo operation in the encryption process is calculated on PSK1 = (P1×MSK1) % (264 + 1). This operation introduces non-linearity into the suggested hybrid cryptographic process. This non-linearity makes it difficult for attackers to apply linear algebraic methods or simple mathematical inversions to break the encryption. The use of a large modulus (264 + 1) ensures a wide range of possible values, making brute-force attacks impractical due to the sheer number of possible outcomes. The addition modulo operation is represented as PSK2 = (P2+MSK2) % 264. The addition modulo operation ensures that the output values are uniformly distributed over the range of possible values. This uniform distribution helps prevent statistical attacks by ensuring there are no patterns or biases in the ciphertext. The bitwise XOR operation is represented as PSK5 = (PSK1 XOR PSK3). The XOR ensures that without knowing the exact keys, reconstructing the original data from the output is infeasible. The complexity of these mathematical operations increases the security of the algorithm, making it difficult to break due to the large modulus operation (264 + 1).

Subsequently, the encrypted data values undergo a second layer of encryption using the DS-RSA algorithm. This algorithm incorporates double exponential operations for encrypting the data values CK2=(C1)PK %RN and C2 =(CK2)FK %RN. This mathematical exponential operation further strengthens the algorithm. It is quite difficult for an attacker to break the data values in this hybrid approach. To decrypt the data values, an attacker would need to know the private key. However, the proposed key generation methodology generates private keys through intricate mathematical techniques, making it difficult for an attacker to discern the actual contents of the private keys. So, the proposed hybrid cryptography approach effectively prevents sensitive data disclosure breaches in the financial sector. This multi-layered encryption strategy ensures that even if one layer is compromised, the overall security of the data remains intact, thus safeguarding sensitive information against unauthorized access. The strategy of the sensitive data disclosure breaches attack is presented in Algorithm 5.

```
Require: M-IDEA and DS-RSA Methodology
Ensure:  CK2 = (C1P K)%RN , C2 = (CK2F K)%RN
input UIDuser , PASSuser
PSK1user  =  EncryptionMIDEA(UIDuser, PASSuser)
PSK2user = EncryptionDSRSA(PSK1user)
if (PSK1user == NULL) then
Print (“No Response from bank database server”)
Exit
else
while (host = mysqlifetchassoc(hostserver)) do
 if   (PSK2user  ==  Bankdatabaseserver ) then
```

```
10: Goto step 16
11: else
12: Exit
13: end if
14: end while
15: end if
16: keyuser = keygeneration ()
17: keyencryption = EncryptionHybrid (keyuser)
18: keydistribution = Distribution (keyencryption )
19: while (SK = mysqlifetchassoc(bankdatabase)) do
20: if  (SK[key] == Keydistribution)   then
21: MRSuser = combine(keyencryption , Keydistribution)
22: if (MRSuser == keyuser) then
23: Flag = 1
24: Exit
25: end if
26: end if
27: end while
28: if (Flag == 1) then
29: Print("No sensitive data disclosure breaches attack")
30: else
31: Print ("Detected sensitive data disclosure breaches")
32: end if
33: Exit
```

Algorithm 5 Strategy of Sensitive Data Disclosure Breaches Attack

6.3. Application in the Financial Sector

The security of the financial sector is paramount due to its critical role in national and global economies and its attractiveness as a target for cybercriminals. It hinges on protecting sensitive data, ensuring the integrity of financial transactions, and maintaining the trust of customers. The proposed hybrid methodology can be used in financial sectors such as banks, provident funds, mutual funds, insurance, etc. It is used to establish secure communication channels between different entities in the financial sector.

E-banking security is a critical aspect of the financial industry, as online banking transactions involve the handling of sensitive information and monetary assets. Ensuring the security of e-banking systems is essential to protect customers, maintain trust, and comply with regulatory requirements.

The hybrid technique is used for secure login processes, ensuring that only authorized individuals have access to sensitive banking information and services. The key generation techniques of DS-RSA are used during the establishment of a secure communication channel. The server's public key encrypts a symmetric session key, which is then decrypted by the client using its private key. Subsequent communication uses the symmetric key for efficiency. The real data that is sent between the client and the server can be encrypted by using M-IDEA. During the key exchange, a shared symmetric key is created and used to encrypt each communication or transaction.

The suggested hybrid methodology using DS-RSA and M-IDEA can be employed to encrypt and secure data stored in the database server. This includes customer information, financial records, and other sensitive data, protecting it from unauthorized access or data breaches. The proposed DS-RSA can be integrated into the encryption and authentication processes of financial transactions. This ensures the integrity and confidentiality of the transaction details, safeguarding against fraudulent activities   and unauthorized access.

The security strategy framework, illustrated in Figure 8 forms a robust shield for the proposed methodology within the banking sector, effectively thwarting cyber threats. Specifically tailored for the banking sector, this framework serves as a robust deterrent against cyber security attacks, enhancing overall security. Within this defense model, a systematic approach is adopted, aiming to identify potential risks and subsequently implement effective countermeasures.

The data values undergo a dual-layer encryption process: first utilizing the M-IDEA with a 512-bit key size methodology and subsequently employing DS-RSA. This dual encryption significantly bolsters the security of the proposed methodology, making it exceedingly challenging for an attacker to track or compromise sensitive data values.

The hybrid decryption process, a core component of our methodology, efficiently restores the original data, maintaining the integrity of information. A noteworthy characteristic of the suggested methodology is its incorporation of advanced mathematical principles and complexity, strategically designed to fortify the banking sector against cybersecurity threats.

This security defense framework is deemed indispensable, ensuring its adaptability to the dynamic landscape of evolving security threats and the continual advancement of attack techniques. This proactive approach ensures that proposed methodology remains at the forefront of safeguarding against emerging cyber risks in the banking sector.

RESEARCH ARTICLE

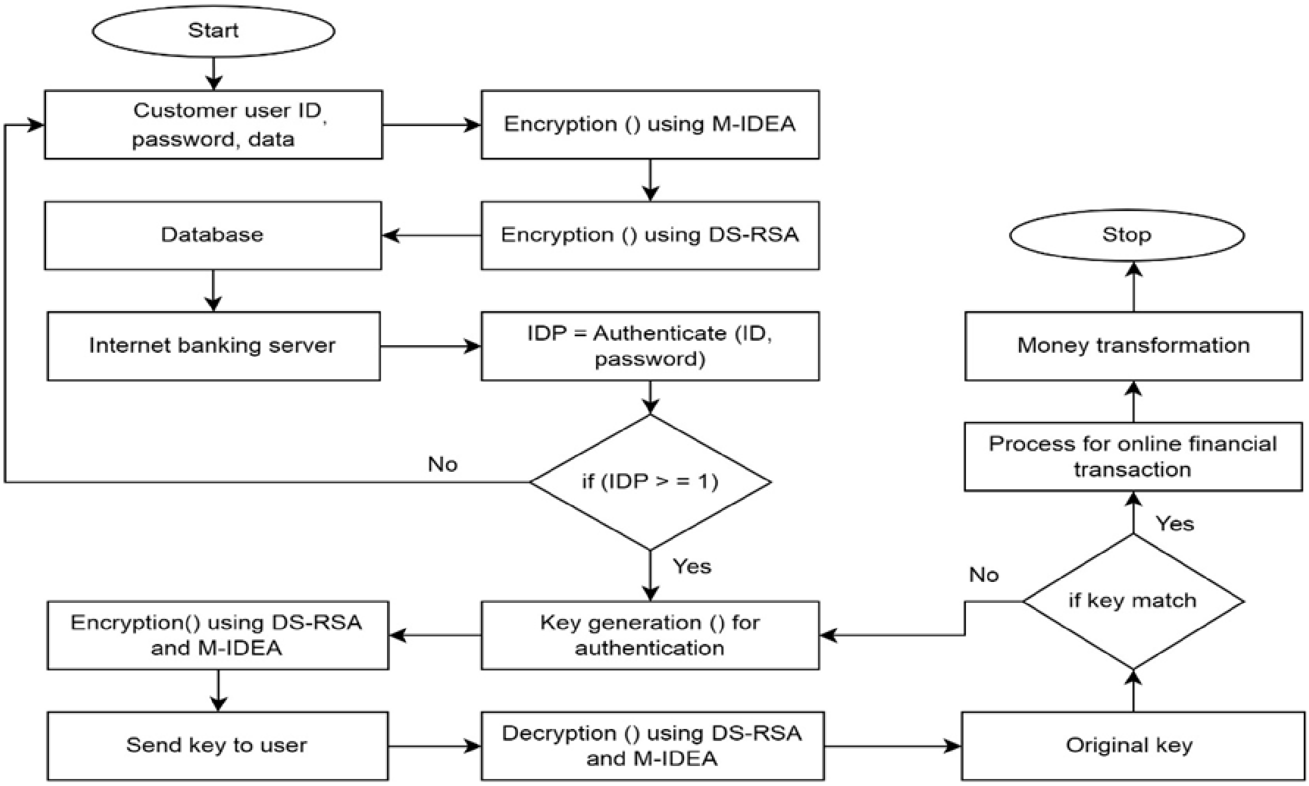
Start
Customer user ID, password, data
Encryption () using M-IDEA
Encryption () using DS-RSA
Database
Internet banking server
IDP = Authenticate (ID, password)
if (IDP > = 1)
No
Yes
Key generation () for authentication
Encryption() using DS-RSA and M-IDEA
Send key to user
Decryption () using DS-RSA and M-IDEA
Original key
if key match
No
Yes
Process for online financial transaction
Money transformation
Stop

Figure 8 Security Strategy Framework in the Financial Sector

7. CONCLUSION

This work has systematically defined the challenges associated with online transactions and put forth a solution by using the hybrid cryptography technique. The research work introduces an enhanced and modified hybrid cryptography technique, combining the strengths of M-IDEA with a 512-bit key size and DS-RSA techniques. A novel key generation methodology is presented that offers an innovative approach to bolstering security in electronic banking. While the proposed method demonstrated commendable performance in the conducted tests, it is acknowledged that the domain of online transactions holds vast potential for substantial advancements. The research introduces robust encryption and decryption techniques to improve the security of financial data transformation. The results of the hybrid methodology demonstrate significantly higher decryption times across all data sizes, starting from 6.01937 seconds for the smallest file and increasing to 58.88446 seconds for the largest file.

The outcome of the comparison ensures that the suggested methodology improved security than traditional RSA techniques. This stark increase indicates a highly complex decryption process, likely integrating multiple security measures to achieve a superior level of data protection. Additionally, the proposed approach is used in the financial sector to achieve robust security of data transformation. The proposed security defense strategy of the Sensitive Data Disclosure Breaches Attack provided robust security in the financial sector. However, longer decryption times are indicative of enhanced algorithmic security, but they also imply a potential decrease in system efficiency and performance particularly in time-sensitive applications. This could impact user experience, especially in scenarios where quick access to encrypted data is crucial.

To achieve a more balanced approach to security and performance the future research work involves incorporating refined versions of the proposed hybrid cryptography decryption technique. Furthermore, this work focuses on developing adaptive security measures techniques based on the current threat level that offers a more user-friendly approach in time-sensitive applications.

REFERENCES

[1] Cayabyab, G.T., Sison, A.M., Medina, R.P.: A secure key scheduling operation for international data encryption algorithm using serpent key schedule operation. In: Proceedings of the 2nd International Conference on Computing and Big Data, pp. 63–67 (2019).

[2] Durga, R., Sudhakar, P.: Implementing rsa algorithm for network security using dual prime secure protocol in crypt analysis. International Journal of Advanced Intelligence Paradigms 24(3-4), 355–368 (2023).

[3] Ma, Z.: Comparative study of the optimization of the multi-prime rsa algorithm. In: 2021 International Conference on Signal Processing and Machine Learning (CONF-SPML), pp. 158–162 (2021). IEEE.

[4] Zhou, J., Zeng, X.: Physical-layer secret key generation based on domain- adversarial training of autoencoder for spatial correlated channels. Applied Intelligence 53(5), 5304–5319 (2023).

[5] Iwasokun, G.B., Akinyokun, O.C., Alawode, S.J., Omomule, T.G.: An rsa algorithm for securing financial data on the cloud. Journal of Advances in Mathematics and Computer Science 34(3), 1–11 (2019).

[6] Venkatesh, G., Gopal, S.V., Meduri, M., Sindhu, C.: Application of session login and one time password in fund transfer system using rsa algorithm. In: 2017 International Conference of Electronics, Communication and Aerospace Technology (ICECA), vol. 2, pp. 732–738 (2017). IEEE.

[7] Dawson, J.K., Twum, F., Hayfron-Acquah, J.B., Missah, Y.M., Ayawli, B.B.K.: An enhanced rsa algorithm using gaussian interpolation formula. International Journal of Computer Aided Engineering and Technology 16(4), 534–552 (2022).

[8] Abdullah, D., Rahim, R., Utama Siahaan, A.P., Ulva, A.F., Fitri, Z., Malahayati, M., Harun, H.: Super-encryption cryptography with idea and wake algorithm. In: Journal of Physics: Conference Series, vol. 1019, p. 012039 (2018).

[9] Aparicio, A., Mart´ınez-Gonz´alez, M.M., Carden˜oso-Payo, V.: App-based detection of vulnerable implementations of otp sms apis in the banking sector. Wireless Networks, 1–14 (2023).

[10] Khan, H.U., Sohail, M., Nazir, S., Hussain, T., Shah, B., Ali, F.: Role of authentication factors in fin-tech mobile transaction security. Journal of Big Data 10(1), 138 (2023).

[11] Guly´as, O., Kiss, G.: Impact of cyber-attacks on the financial institutions. Procedia Computer Science 219, 84–90 (2023).

[12] Tso, R.: Untraceable and anonymous mobile payment scheme based on near field communication. Symmetry 10(12), 685 (2018).

[13] Khan, H.U., Malik, M.Z., Nazir, S., Khan, F.: Utilizing bio metric system for enhancing cyber security in banking sector: a systematic analysis. IEEE Access (2023).

[14] Al-Alawi, A.I., Al-Bassam, M.S.A.: The significance of cybersecurity system in helping managing risk in banking and financial sector. Journal of Xidian University 14(7), 1523–1536 (2020).

[15] Shanthakumari, R., Malliga, S.: Dual-layer security of image steganography based on idea and lsbg algorithm in the cloud environment. S¯adhan¯a 44, 1–12 (2019).

[16] Sarna, S., Czerwinski, R.: Small prime divisors attack and countermeasure against the rsa-otp algorithm. Electronics 11(1), 95 (2021).

[17] Kanu, C., Nnam, M.U., Ugwu, J.N., Achilike, N., Adama, L., Uwajumogu, N., Obidike, P.: Frauds and forgeries in banking industry in africa: a content analyses of nigeria deposit insurance corporation annual crime report. Security Journal 36(4), 671–692 (2023).

[18] Khan, H.U., Sohail, M., Nazir, S., Hussain, T., Shah, B., Ali, F.: Role of authentication factors in fin-tech mobile transaction security. Journal of Big Data 10(1), 138 (2023).

[19] Mishra, S.: Exploring the impact of ai-based cyber security financial sector management. Applied Sciences 13(10), 5875 (2023).

[20] Shivaramakrishna, D., Nagaratna, M.: A novel hybrid cryptographic framework for secure data storage in cloud computing: Integrating aes-otp and rsa with adap- tive key management and time-limited access control. Alexandria Engineering Journal 84, 275–284 (2023).

[21] Abid, R., Iwendi, C., Javed, A.R., Rizwan, M., Jalil, Z., Anajemba, J.H., Biamba, C.: An optimised homomorphic crt-rsa algorithm for secure and efficient communication. Personal and Ubiquitous Computing 27, 1405–1418 (2023).

[22] Stanikzai, A.Q., Shah, M.A.: Evaluation of cyber security threats in banking systems. In: 2021 IEEE Symposium Series on Computational Intelligence (SSCI), pp. 1–4 (2021). IEEE.

[23] Ghelani, D., Hua, T.K., Koduru, S.K.R.: Cyber security threats, vulnerabilities, and security solutions models in banking. Authorea Preprints (2022).

[24] Kumar, M., et al.: An overview of cyber security in digital banking sector. East Asian Journal of Multidisciplinary Research 2(1), 43–52 (2023).

[25] Ganavi, M., Prabhudeva, S., Nayak, S.N.: A secure image encryption and embed- ding approach using mrsa and rc6 with dct transformation. International Journal of Computer Networks and Applications (IJCNA) 9(3), 262–278 (2022).

[26] Kanda, G., Ryoo, K.: Vedic multiplier-based international data encryption algorithm crypto-core for efficient hardware multiphase encryption design. Webology 19(1) (2022).

[27] Neela, K., Kavitha, V.: An improved rsa technique with efficient data integrity verification for outsourcing database in cloud. Wireless Personal Communications 123(3), 2431–2448 (2022).

[28] Kuppuswamy, P., Al, S.Q.Y.A.K., John, R., Haseebuddin, M., Meeran, A.A.S., et al.: A hybrid encryption system for communication and financial transactions using rsa and a novel symmetric key algorithm. Bulletin of Electrical Engineering and Informatics 12(2), 1148–1158 (2023).

[29] Maria, R., Anitha, V.: Light weight asymmetric cryptographic algorithm for financial transactions through mobile application. International Journal of Computer Applications 975, 8887.

[30] Iyer, S.C., Sedamkar, R., Gupta, S.: A novel idea on multimedia encryption using hybrid crypto approach. Procedia Computer Science 79, 293–298 (2016).

[31] Mangalagowri, R., Venkataraman, R.: Hypervisor attack detection using advanced encryption standard (hadaes) algorithm on cloud data. International Journal of Computer Networks and Applications, 9 (5) 555–567 (2022).

[32] Hermawan, N., Winarko, E., Ashari, A.: Eight prime numbers of modified rsa algorithm method for more secure single board computer implementation. International Journal on Advanced Science, Engineering and Information Technology 11(6), 2375–2384 (2021).

[33] Wardak, O., Sinha, D., Sethi, A.: Encryption and decryption of signed graph matrices through rsa algorithm. Indian Journal of Pure and Applied Mathematics, 1–8 (2023).

[34] Karim, R., Rumi, L.S., Ashiqul Islam, M., Kobita, A.A., Tabassum, T., Sagar Hossen, M.: Digital signature authentication for a bank using asymmetric key cryptography algorithm and token based encryption. In: Evolutionary Computing and Mobile Sustainable Networks: Proceedings of ICECMSN 2020, pp. 853–859 (2021). Springer.

[35] Kaur, D., et al.: Efficient encryption mechanism for financial transactions: Avoiding data loss and tackling collisions. Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12(11), 1861–1872 (2021).

[36] Merkepci, M., Abobala, M., Allouf, A.: The applications of fusion neutrosophic number theory in public key cryptography and the improvement of rsa algorithm. Fusion: Practice and Applications (2023).

[37] Alatawi, M.N.: A hybrid cryptography and logixgboost model for intelligent and privacy protection in wireless body sensor networks (wbsns). International Journal of Computer Networks and Applications, 10 (2) 166–179 (2023).

[38] Joseph, M., Mohan, G.: Design a hybrid optimization and homomorphic encryption for securing data in a cloud environment. International Journal of Computer Networks and Applications (IJCNA) 9(4), 387–395 (2022).

[39] Alatawi, M.N.: A hybrid cryptographic cipher solution for secure communication in smart cities. International Journal of Computer Networks and Applications, 10(5) 776 - 791 (2023).

[40] Kbar, G., Mansoor, W.: Modified rsa using triple keys based encryption/decryption. Jordan Journal of Electrical Engineering. All rights reserved-Volume 7(1), 2 (2021).

[41] Haldar, B., Mukherjee, P. K. & Saha, H. N , An Approach of Modified IDEA with 1024 Bits Key to Enhance Security and Efficiency of Data Transmission in the Healthcare Sector. International Journal of

RESEARCH ARTICLE

Mathematical, Engineering and Management Sciences, 9(6), 1453-1482, (2024).

Authors

Bilas Haldar working as an Assistant Professor in the Department of Computer Science and Engineering at The Neotia University. He is pursuing a Ph.D. in Computer Science & Engineering at The Neotia University. Mr. Haldar obtained his M.Tech in Software Engineering from Maulana Abul Kalam Azad University of Technology. He has more than 11 years of academic, teaching, and research experience. His research interests include Cyber Security, Cryptography, Network Security, Data Security, Application Security and Machine Learning.

Prof. (Dr.) Partha Kumar Mukherjee has around 29 years of industrial and research experience in the field of IT and ITeS. He has expertise in the domain area of Digital Governance, Investment Banking, Insurance, Healthcare, Education, Aerospace, and Retail. He is working as a Dean of the School of Science and Technology and Professor in the Department of Computer Science and Engineering at The Neotia University. Previously, he was engaged with National eGovernance Division (NeGD), MeitY, Government of India as a Principal Consultant/ Head of the State SeMT (State eGovernance Mission Teams) Team, and he was deputed at Directorate of Information Technology (DIT), Government of Tripura as a SeMT Head. He was also engaged with the HCL Technologies for 14 years as a Program Manager, Projects delivery group. At HCL, he was engaged in a leadership role in the Retail and BFSI – Business Intelligence and Data Science Group. His core expertise is in the area of Digital Government, Block Chain, Business Intelligence System, Big Data Analytic and Data Science. He has done his PhD (Engg.) in the area of Data Streaming, Data Analytic in Multimedia Database Management System from CSE department, Jadavpur University. He has global work experience under multiple verticals with countries like, UK, USA, Asia Pacific Countries, and Europe.

Dr. Himadri Nath Saha is the Head of the Department of Computer Science at SNEC, Calcutta University, Kolkata. He has completed his Bachelor of Engineering from Jadavpur University, his Master of Engineering from Indian Institute of Engineering, Science and Technology(IIEST, Shibpur and has obtainedhis Ph.D. in Engineering from Jadavpur University. His research interest includes Machine Learning, IOT, Wireless Communication, Mobile Ad-hoc Networks, Cyber Security, Network Security, Cryptography and Algorithms. He has several publications in international journals and has collaborated with multiple researchers on various projects. He has also written a textbook on “Database Management System”. Dr. Saha is the "Fellow" of many premiere organisations like Institution of Engineers India(IEI) and The Institution of Electronics and Telecommunications Engineers(IETE), India and senior member of IEEE(USA). He is the Branch Counselor of IEEE Student Branch & Faculty Head of ACM Student Chapter.

How to cite this article:

Bilas Haldar, Partha Kumar Mukherjee, Himadri Nath Saha, “A Robust Security Strategy Using Hybrid Cryptography Approach to Protect Data in the Financial Sector”, International Journal of Computer Networks and Applications (IJCNA), 11(6), PP: 749-773, 2024, DOI: 10.22247/ijcna/2024/46.