Scientific Research Publishing

Modern Economy, 2023, 14, 1192-1217
https://www.scirp.org/journal/me
ISSN Online: 2152-7261
ISSN Print: 2152-7245

Review of Algorithms for Securing Data Transmission in Mobile Banking

Daniel Okari Orucho[1*], Fredrick Mzee Awuor[1], Ratemo Makiya[2], Collins Oduor[3]

[1]Department of Computing Sciences, School of Information Science and Technology, Kisii University, Kisii, Kenya
[2]Department of Computing and Information Technology, School of Pure and Applied Sciences, Mama Ngina University, Gatundu, Kenya
[3]School of Science and Technology, United States International University-Africa, Nairobi, Kenya
Email: *danielokari@yahoo.com

How to cite this paper: Orucho, D. O., Awuor, F. M., Makiya, R., & Oduor, C. (2023). Review of Algorithms for Securing Data Transmission in Mobile Banking. *Modern Economy, 14,* 1192-1217.
https://doi.org/10.4236/me.2023.149062

Received: June 28, 2023
Accepted: September 12, 2023
Published: September 15, 2023

Abstract

With the dawn of mobile banking applications, bank customers can now register for M-banking and download applications that aid them to access services from the bank server remotely from their mobile devices. The conversation between the bank's server and the client application requires a secure connection. However, M-banking is often conducted via unsecure wireless networks on which adversaries can use available techniques to hack into system to steal sensitive financial information including money. This paper's objective is to review the state-of-the-art algorithms that secure data on transit in M-banking. Thus, we document the strengths and weaknesses of these mechanisms and report on structure of their operation. The study reviewed various encryption algorithms such as Rivest-Shamir-Adleman Algorithm, Elliptic Curve Cryptography, Digital Signature Algorithm, Blowfish algorithm, Advanced Encryption Standard, Data Encryption Standard and Triple Data Encryption Standard. In addition, the study reviewed steganography and hybrid algorithms. From this study, we show that Advanced Encryption Standard is the most preferred standard for M-banking because there are no specific attacks against it so far. However, since technology is changing fast, Advanced Encryption Standard might not provide security in M-banking for long. Therefore, this study shows and recommends utilization of a combination of Advanced Encryption Standard algorithm and Least Significant Bit steganography to produce a robust hybrid algorithm that is tamperproof from flaws existing in current cryptosystems.

Keywords

Cryptography, Steganography, Hybrid Algorithms, Encryption, Decryption

1. Introduction

Applications used for mobile banking (M-banking) provide users the flexibility to communicate with financial institutions over the internet at any time using a mobile device (Sakala & Phiri, 2019). Banks have deployed platforms such as M-banking applications (so called apps) that allow their customers to access their bank accounts remotely. Customers can consequently access financial services like monitoring account balances, sending money, and selling stocks using M-banking (Nawaz, Motiwalla, & Deokar, 2018). Bookings, loan payments, and airtime top-ups are other additional services that M-banking provides. M-banking benefits both banking institutions that offer such services and bank customers because it allows secure 24/7 remote access to services without the need to physically visit a bank branch (Sethi & Acharya, 2018). M-banking also lowers operating costs and boosts competitiveness (Purohit & Arora, 2021).

Mobile devices and wireless networks are used to transmit sensitive financial data, such as transactions and account information (personal identification numbers, usernames, and passwords), to access banking services remotely. However, a number of threats pose a risk to undermine the security of M-banking (McDonald, 2020). Eavesdropping, malware, phishing, denial of service, and unauthorized access are some of the most frequent attacks on mobile banking systems (Falade & Ogundele, 2022). Majority of these Attacks take place on insecure networks (Lula, Dospinescu, Homocianu, & Sireteanu, 2021). In order to protect customer's accounts from unauthorized access, there are several techniques that can be utilized such as encryption algorithms and steganography (Joshi, Gill, & Yadav, 2018).

This paper reviewed different types of encryption algorithms, steganography and hybrid algorithms that can be utilized to secure user data on transit in M-banking. From the review, AES algorithm is the most recommended and secure algorithm that can be utilized for M-banking because it is quick in encryption and decryption and has not yet been compromised. However, since technology advances rapidly, AES might not continue enjoying its security tenet. Due to this reason, this paper finds that a combination of AES encryption with steganography algorithm such as Least Significant Bit (LSB) provides an extra layer of security. This is because AES algorithm contributes security attributes such as encryption and decryption using a key while LSB steganography contributes security attribute of embedding a message in an image. Thus the hybrid algorithm formed is tamperproof such that even if an adversary discovers the presence of a message in the image using available software, it will be difficult to know the encryption key used in order to decrypt the message.

This study enlightens banks offering M-banking on the need to develop secure systems with enhanced security that is tamperproof for user data on transit. This will mitigate cybercriminals from accessing customers' bank accounts to steal confidential information and money. Knowledge from this paper is crucial for decision making when it comes to developing of M-banking applications that are utilized by customers to access banking services remotely.

This paper reviewed the following encryption algorithms: Rivest-Shamir-Adleman (RSA), Elliptic Curve Cryptography (ECC), Digital Signature Algorithm (DSA), BlowFish Algorithm, Advanced Encryption Algorithm (AES), Data Encryption Standard (DES), and Triple Data Encryption Standard (3DES), steganography, and hybrid algorithms. The remaining part of this review is structured in the following manner: definition of the problem is presented in Section 2, followed by methods and materials in Section 3, algorithms for securing data transmission in M-banking is discussed in Section 4, Section 5 contains results, discussion, recommendation and conclusion.

2. Problem Definition

In this paper we provide a review of encryption algorithms, steganographic algorithms and hybrid algorithms that can be utilized to secure user data transmission in M-banking. Numerous algorithms have been designed to ensure secure data transmission in M-banking. However, there are no effective security measures that can address new security threats. The currently proposed algorithms, while promising, still need to be improved in order to defend against various types of attacks, including mobile malware, spoofing, social engineering, and man-in-the-middle (mitm) attacks.

Strong algorithms that ensure protection of customers' private information must be developed in order to protect bank servers and clients from adversaries that try to steal money and get access to confidential data using illegal methods. Therefore, there is need to design robust algorithms that can be used to secure data transmission in M-banking. The objective of this paper is to review the state-of-the-art algorithms that secure data transmission in M-banking. Thus, we document the strengths and weaknesses of these mechanisms and report on structure of their operation.

3. Methods and Materials

This review examines existing methods and techniques of encryption algorithms, steganography, and hybrid algorithms researched between 2012 and 2023. This section includes subsections on data sources, search processes, data selection, and data extraction.

The primary sources for this review have been selected from different scientific databases such as Institute of Electrical and Electronics Engineers Xplore Digital Library (IEEE Xplore), Science Direct (https://www.sciencedirect.com/), Springer Link (https://link.springer.com/), Google Scholar (https://scholar.google.com/), Association for Computing Machinery Ditigal Library (https://dl.acm.org/), and peer reviewed international journals.

The search process was carried to identify potential research papers from scientific databases using pre-selected search keywords or strings including algorithms for securing data transmission in M-banking, steganography, and hybrid algorithms. Additionally, the following Boolean operators have been used in the search process: ((Cryprography in mobile banking) OR (Analysis of cryptogra-

phy in mobile banking)), (((Symmetric key cryptography in mobile banking) OR (Symmetric key algorithms in mobile banking)), and (cryptography AND (image steganography)) OR (encryption AND (image steganography)) AND (spatial domain).

Data selection was done through filtering results obtained based on keywords in English language. The criteria employed scanned whether obtained results discusses about cryptographic algorithms, if the research articles mentioned concepts about steganography, and finally if the articles discuss about hybrid algorithms. In order to apply inclusion and exclusion criteria, we put down exclusion criteria such as duplicate papers, full-text availability, and papers that are not related to algorithms securing data transmission in M-banking. Approximately 200 publications were discovered after this work was completed, and those that were pertinent were chosen based on the search criteria. In the end, 50 related studies were found and used in this review.

4. Algorithms for Securing Data Transmission in Mobile Banking

Data security is important in modern communication systems. There are several techniques that can be employed to strengthen data security such as utilization of encryption of data on transit and in storage, steganography, and hybrid algorithms. The following section discusses the different types of algorithms that are used for securing data transmission in M-banking.

4.1. Cryptographic Algorithms

Cryptographic algorithms employ a technique of converting plaintext into unreadable format to thwart efforts of adversaries to intercept text being sent. At the receiver side, unreadable format of text is transformed into plaintext using encryption algorithm and a key. The two categories of encryption modes are symmetric in which encryption and decryption utilizes the same key, and asymmetrical whereby two different keys are used for the process of encryption and decryption. The three functions of cryptographic algorithms are: encoding whereby original plaintext is rendered unreadable, key which is used in performing the process of encryption and decryption, and finally decryption, whereby unreadable text is transformed into readable text (Sari, Rachmawanto, & Sari, 2017). The various categories of cryptographic algorithms are covered in detail in the following section.

4.1.1. Rivest-Shamir-Adleman Algorithm

This is asymmetric cryptosystem wherein two different keys are utilized to encrypt and decrypt data (Bhanot & Hans, 2015). Rivest-Shamir-Adleman (RSA) is widely used in networks to secure data. The integer factorization issue and the RSA problem, such as determining the Nth root, are the two main unsolved RSA puzzles in which the result of two prime numbers is N. According to number theory, while factorization is challenging, calculating the product of two prime

integers is straightforward. The principle of huge numbers is crucial to RSA security. Factorization is difficult due to the key size range of the RSA algorithm, which is between 2048 and 4096. The variables d and N, where d stands for the decryption key, are the foundation for the decryption of the RSA method. RSA has a time complexity of O (n^2). This cryptosystem incorporates: key generation, encryption, and decryption. **Table 1** (Saini & Vandana, 2022) illustrates RSA key generation procedure, while **Table 2** illustrates RSA encryption and decryption process.

A one-way function produced by a modular exponential function in the multiplication groups (p, x) and (q, x), where p, q is a prime number and $n = (p \times q)$ is utilized in encrypting and decrypting procedures of the cryptographic system where p and q is a prime number and $\emptyset(n) = (p-1) \times (q-1)$. $d_n d_{\varnothing n}$ (Meng & Zeng, 2015). The fundamental prerequisite for using the encryption method is the construction of a key (which is public) such that for encryption to be successful, plain text should be larger than 0 and less than the modulus (n) value of public key. It is essential to reshape plaintext into American Standard Code for Information Interchange (ASCII) code as part of the encryption process if it comprises letters or symbols. This allows for the growth of the RSA encryption process (Anada, Yasuda, Kawamoto, Weng, & Sakurai, 2019).

The plaintext must first be transformed into decimal integers in RSA encryption before performing the multiplication using the algorithm above. When converting plaintext to decimal numbers, one must consider the ASCII code value. The quantity of the key-size used, and the amount of text to be encrypted determines how long the encryption process takes for RSA algorithm (Thiyagarajan & Meenakshi, 2019). The RSA algorithm's decryption procedure resembles encryption in several ways. The application of values e and d differs. Using a private key, this decryption method also uses a modular and exponential algorithm and a decimal number is the formula's result (Bunder, Nitaj, Susilo, & Tonien, 2017).

The ASCII code values are used to convert plaintext to decimal numbers during the encryption process and vice versa. The decimal value of the plaintext value must be translated to a character utilizing the ASCII value code (Seo, 2020).

Table 1. RSA key generation process (Tahir, 2015).

	Key Generation Procedure
1	Generate two large prime numbers p and q. € greatest common divisor (gcd) (p, q) = 1
2	Compute $n = p * q$
3	Compute Euler's Totient Function $\emptyset(n) = (p-1) * (q-1)$ where ø = totient
4	Choose public key integer e, where $1 < e < \emptyset(n)$ € $\gcd(\emptyset(n), e) = 1$
5	Compute private key integer d, $d = e - 1 \bmod \emptyset(n)$
6	Public key is (e, n) and private-key is (d, n)

RSA is a contemporary encryption cryptosystem that uses expressions with exponential functions (Luy, Karatas, & Ergin, 2016). Plaintext is encoded in blocks, each with a binary value below a particular threshold (n). Table 2 provides a description of encryption and decryption procedures for plain P block and C cipher (Lin, Sun, & Qu, 2018).

Table 2. RSA encryption and decryption process (Lin, Sun, & Qu, 2018).

	Encryption Process		Decryption Process
1	$C = Pe \bmod n$	1	$P = Cd \bmod n$

Table 2 demonstrates encoding process algorithm with following parameter denotations: c = cipher, P = Plain, e = Public key and n = product of two prime numbers. The same parameters are used in the decoding process. The value of n must be understood by both the sender and the recipient. Only the recipient is aware of the value of d, whereas the sender is aware of e's value. In light of this, it may be said that this algorithm's key is represented by the formula KU = e, n, and its private key is represented by KR = d, n. (Anada, Yasuda, Kawamoto, Weng, & Sakurai, 2019; Sharma, Agrawal, Pandey, Khan, & Dinkar, 2022). Confidentiality, secrecy, authentication, integrity, and non-repudiation are the major security services that RSA offers (Sharma & Bohra, 2017). It is challenging for hackers to crack the algorithm because of its superior security public-key cryptosystem (Gaur, Mehra, & Kumar, 2018). However, RSA is slower than other cryptosystems because it requires key deposits, is vulnerable to brute force assaults and timing attacks (Mitra, Jana, Bhattacharya, Pal, & Poray, 2017). RSA algorithm should not be used in numerous systems, including M-banking, as a result of these flaws (Jahan, Asif, & Rozario, 2015).

4.1.2. Elliptic Curve Cryptography

Miller and Koblitz created asymmetric key cryptosystem known as elliptic curve cryptography (ECC) (Liu, Huang, Hu, Khan, & JeongSeo, & Zhou, 2017; Hsiao, 2017). Due to its tiny key size and high network speed, ECC is a desirable alternative for devices with limited resources and has grown in popularity as a security option in recent years. AES and ECC are two encryption algorithms that are used by a number of technologies and protocols. ECC is used in Bluetooth Low Version and the limited application protocol (Granjal, Monteiro, & Silva, 2015).

The equation for elliptic curve is used to create keys. ECC gets its security from the size of Elliptic Curve (EC) and a Discrete Logarithmic (DL) framework that is more challenging to solve than factoring (Bhaskar & Mohan, 2019). Even though ECC executes slower than AES, it can nevertheless offer security services such as non-repudiation, authentication, and confidentiality. Security of ECDL problem lies on its hardness and is deemed as the foundation stone for security of the algorithm. If the attacker is successful in obtaining both n and n × m, he must now determine the value of m which is a challenging task (Chande, Lee, & Li, 2018).

The most common attacks on ECC include random walks-based attacks, random walks with special conditions, and attacks based on multiplicative groups. Additional assaults on ECC based protocols are: side channel attacks, power analysis attacks, electromagnetic analysis attacks, error message analysis, fault analysis, and timing attacks. This form of assaults on ECC can be mitigated through software and hardware-based techniques. For software-based solutions, randomness or dummy operations on algorithms, for instance, can be used. Metal layers are utilized in hardware-based fixes. Key for every message can also be used in timing assaults (Chande, Lee, & Li, 2018). Additionally, ECC is slower than most symmetric algorithms and therefore not suitable to be utilized in M-banking.

4.1.3. Digital Signature Algorithm

This method is made possible by a hashing algorithm, in which even a small modification to data affects outcome in another hash or digest. A changed piece of data generates a new hash, alerting the receiver that the data they just got was altered or corrupted in transit and wasn't delivered by the intended sender (Thapar & Sarangal, 2018). Security during data transmissions is further enforced by digital signatures. Digital signatures are utilized to accomplish authentication, integrity, and non-repudiation. Three steps that make up the Digital Signature Algorithm (DSA) process are key creation, signature creation, and signature verification (Simplilearn, 2022).

The DSA method is advantageous since key generation is more rapid and more robust in terms of security and stability than using RSA method, it uses less storage space during its whole cycle, and is patent-free, allowing for its unrestricted use worldwide (Simplilearn, 2022). The discrete logarithm issue must first be solved for the digital signature algorithm to be broken, making it a secure digital signature scheme (Stinson, 2006).

Key-only attacks, known-message attacks, chosen-message attacks, and forgeries are the most frequent DSA assaults. The attacker's only tool in key-only attacks is the public verification key. In known-message arrack, the attacker is provided with a valid signature for a number of messages that they are aware of but have not specifically chosen. The attacker initially ascertains the signatures on any random messages selected by the attacker in a chosen-message attack (Kumar, Reddy, Rinaldi et al., 2021). Two different kinds of forgery attacks are potential forgeries and selective forgeries. Existential forgery occurs when a third party produces a message/signature pair (m) that was not produced by the authorized signer. During selective forging, the adversary creates a message/signature pair (m), where m has been pre-decided by the adversary (Kumar, Reddy, Rinaldi et al., 2021). Due to slow speed and flaws in signature verification, DSA is not suitable for M-banking.

4.1.4. BlowFish Algorithm

Bruce Schneier made the discovery of the Blowfish algorithm in 1993 (Anwar,

Hasan, Hasan, Loren, & Hossain, 2019). Data encryption and decryption using the Blowfish algorithm utilize the identical secret keys in this symmetric block cipher. Block ciphers separate messages into blocks of a certain size for encryption and decoding. As a result, communications with lengths that are not multiples of 8 bytes must be padded. The block size of Blowfish is 64 bits (Valmik & Kshirsagar, 2014). Key size of Blowfish algorithm ranges from 32 to 48 bits. There are also variations with 14 rounds or fewer (Kumar, Thakur, & Kalia, 2011). It takes up to 5 Kilobytes of memory and is regarded as the quickest block cipher that has been developed over the years to allow everyone to utilize encryption without worrying about copyright and patent issues.

The key-expansion and data encryption are divided into two separate parts. A condition with a maximum length of 448 bits is expanded into a variety of smaller keys combined 4168 bytes. The most prevalent method of data encryption uses a 16-round configuration. Every round relies on key arrangement, and the substitution is done using same information. All of these operations are carried out by applying XORs and adds to 32-bit words. Along with the aforementioned actions, a data lookup using four indexed arrays is also performed (Nie & Zhang, 2009).

Because the keys do not change regularly, Blowfish algorithm is appropriate for application areas including database security and internet commerce. In cases where massive data caches are taken into account on a 32-bit microprocessor, the Blowfish algorithm performs better than the majority of other algorithms. Blowfish is a somewhat a rapid block cipher due to the round's simplicity and low number of rounds (CommonLounge, 2018). The main schedule in Blowfish is a little tiresome. Compared to AES's 128 bits, Blowfish's 64-bit short block size is more vulnerable to assaults. However, because two people have the same key, Blowfish method is unable to provide confirmation and non-denial.

4.1.5. Advanced Encryption Standard

Electronic data can be secured using Advanced Encryption Standard (AES) cryptographic method, authorized by Federal Processing Standards Publications (FIPS). AES has the capacity to both encrypt and decode digital data. Data blocks of 128 bits can be encrypted and decrypted using keys with bit lengths of 128, 192, and 256 (FIPS 197, 2023). Internally, the state structure of AES has two dimensions of (4 by 4) matrix arrangement bytes on which advanced AES block ciphers design is run.

Each individual byte in the state array, designated as s, contains indicators: row of token r which ranges $0 \le r < 4$ and column token c which ranges $0 \le c < 4$. Similarly, a state-specific byte is represented using the following: $s_{r,c}$ or $s[r, c]$. The move in the stipulations of AES is to replicate input arrangement of bytes represented as in to the four-by-four square variable values s illustrated in Equation 1.

$$s[r,c] = in[r+4c] \quad \text{for} \quad 0 \le r < 4 \quad \text{and} \quad 0 \le c < 4 \tag{1}$$

A sequence of transformations is then applied to the state array, after which its final value is copied to the output array of bytes $out_0, out_1, \cdots, out_{15}$ as illustrated in Equation 2.

$$out[r+4c] = s[r,c] \quad \text{for} \quad 0 \le r < 4 \quad \text{and} \quad 0 \le c < 4 \tag{2}$$

The general function for executing AES-128, AES-192, or AES-256 is denoted by CIPHER (); its inverse is denoted by INVCIPHER ().The core of the algorithms for CIPHER () and INVCIPHER () is a sequence of fixed transformations of the state called a round. Each round requires an additional input called the round key; the round key is a block that is usually represented as a sequence of four words (such as 16 bytes).

An expansion routine, denoted by KEYEXPANSION (), takes the block cipher key as input and generates the round keys as output. In particular, the input to KEYEXPANSION () is represented as an array of words, denoted by key, and the output is an expanded array of words, denoted by *w*, called the *key schedule*.

The block ciphers AES-128, AES-192, and AES-256 differ in three respects: 1) the length of the key; 2) the number of rounds, which determines the size of the required key schedule; and 3) the specification of the recursion within KEYEXPANSION (). For each algorithm, the number of rounds is denoted by *Nr*, and the number of words of the key is denoted by *Nk*. (The number of words in the state is denoted by *Nb* for Rijndael in general; in this Standard, *Nb* = 4.) The specifc values of *Nk*, *Nb*, and *Nr* are given in Table 3. No other configurations of Rijndael conform to this Standard.

Table 3. Key-BLOCK-ROUND COMBINAtions (FIPS 197, 2023).

	Key length		Block size		Number of rounds
	Nk	in bits	*Nb*	(in bits)	*Nr*
AES-128	4	128	4	128	10
AES-192	6	192	4	128	12
AES-256	8	256	4	128	14

The three inputs to CIPHER() are: 1) the data input *in*, which is a block represented as a linear array of 16 bytes; 2) the number of rounds *Nr* for the instance; and 3) the round keys. Thus,

$$\begin{aligned} &\text{AES-128 } (in, key) = \text{Cipher } (in, 10, \text{KEYEXPANSION } (key)) \\ &\text{AES-128 } (in, key) = \text{Cipher } (in, 12, \text{KEYEXPANSION } (key)) \\ &\text{AES-128 } (in, key) = \text{Cipher } (in, 14, \text{KEYEXPANSION } (key)) \end{aligned} \tag{3}$$

The inverse permutations are defined by replacing CIPHER () with INVCIPHER () in Equation (3).

The rounds in the specification of CIPHER () are composed of the following four byte-oriented transformations on the state: SUBBYTES () which applies a substitu-

tion table (S-box) to each byte, SHIFTROWS () shifts rows of the state array by different offsets, MIXCOLUMNS () mixes the data within each column of the state array and ADDROUNDKEY () combines a round key with the state. The pseudocode in **Table 4** illustrates algorithm formulation for Cipher () (FIPS 197, 2023).

The first step in line 2 is to copy the input into the state array. After an initial round key addition in line 3, the state array is transformed by Nr applications of the round function in lines 4 to 12; the final round in lines 10 - 12 differs in that the MIXCOLUMNS () transformation is omitted. The final state is then returned as the output in line 13.

To generate $4 \times (Nr + 1)$ words from a key, the KEYEXPANSION () routine is used. For instance, four words are produced for each of the Nr +1 ADDROUNDKEY () applications found in the Cipher specification (). The routine's output is a list of words in order designated $w[i]$, where i falls in the vicinity of $0 \le i < 4 \times (Nr + 1)$. Then fixed phrases are invoked through KEYEXPANSION (), which is indicated by $Rcon[j]$ for $1 \le j \le 10$. The round constants are ten words. Each of ten round keys for AES-128 is generated by calling a unique round constant. The first six and eight of these constants are referred to by the key expansion algorithm for AES-192 and AES-256, respectively. **Table 5** demonstrates pseudocode for KEYEXPANSION () function as in (FIPS 197, 2023).

To implement INVCIPHER (), the transformations in the specification of CIPHER () are inverted and executed in reverse order. The inverted transformations of the state denoted by INVSHIFTROWS (), INVSUBBYTES (), INVMIXCOLUMNS (), and ADDROUNDKEY () (FIPS 197, 2023).

AES is used in banking systems and government organizations for secure data

Table 4. Pseudocode for CIPHER () (FIPS 197, 2023).

Pseudocode for Cipher () Algorithm	
1	procedure Cipher (*in, w, Nr*)
2	state ← *in*
3	state ← AddRoundKey(*state,w*[0..3])
4	for round from 1 to $Nr - 1$ do
5	state ← SubBytes(*state*)
6	state ← ShiftRows(*state*)
7	state ← MixColumns(*state*)
8	state ← AddRoundKey(*state,w*[4 ∗ *round*..4 ∗ *round* + 3])
9	end for
10	state ← SubBytes(*state*)
11	state ← ShiftRows(*state*)
12	state ← AddRoundKey(*state,w*[4 ∗*Nr*..4 ∗*Nr* + 3])
13	return state
14	end procedure

Table 5. Pseudocode for KEYEXPANSION () (FIPS 197, 2023).

Pseudocode for Key Expansion () Algorithm	
1	procedure Key Expansion (key)
2	i ← 0 3
3	while i ≤ *Nk* −1 do 4
4	*w[i]* ← *key*[4 ∗ i..4 ∗ i + 3]
5	*i* ← i + 1 6
6	end while
7	*while i* ≤ 4 ∗*Nr* + 3 do
8	*temp* ← *w* [*i* − 1]
9	if *i mod Nk* = 0 then
10	*temp* ← SubWord(RotWord(temp)) ⊕ *Rcon*[*i/Nk*]
11	else if *Nk* > 6 and *i mod Nk* = 4 then
12	*temp* ← SubWord(*temp*)
13	end if
14	*w*[*i*] ← *w*[*i*–*Nk*] ⊕ *temp*
15	: *i* ← *i* + 1
16	end while
17	return *w*
18	end procedure

transmission (Khelifi, 2013). This is due to the fact that it might take longer than the universe's age to crack a 128-bit AES key. However, according to (Amrita, Gupta, & Mishra, 2018), AES is open to numerous side channel attacks. These types of attacks utilize the descriptive data gleaned from the protocols and cryptographic primitives' implementation. Timing, power usage and electromagnetic radiation aspects can be used to obtain this characteristic information. Computational errors, variations in frequency or temperature, and hardware or software flaws can all produce additional types of information. Side channel attacks exploit the features of the hardware and software components as well as the cryptographic primitive's implementation structure (Jani, 2015).

Other cryptanalysis attacks against AES include algebraic attacks, cube attacks, eXtended Linearization (XL), eXtended Sparse Linearization (XSL), and collision attacks (Anwar, Hasan, Hasan, Loren, & Hossain, 2019), among others, are advancing steadily but there haven't been any major developments announced yet. The AES won't have the same lifespan as the conventional algorithm suite certified for classified applications given these developments. But there are practical countermeasures that, when used correctly, can eradicate these weaknesses at the equipment level (Amrita, Gupta, & Mishra, 2018). AES and steganography are two examples of combinations of two algorithms that can

be used to bolster security further and circumvent cryptanalysis's weaknesses against AES.

4.1.6. Data Encryption Standard

Data Encryption Standard (DES) is an industry-standard technique for securing computer and telecommunications data. According to (Al-Hazaimeh, Alhindawi, Hayajneh, & Almomani, 2013) DES is a block cipher of the Feistel type in which the left and right sides of a block of bits are processed individually across a number of rounds. It's interesting that a Feistel encryption can be inverted under the condition that the function (f) used to operate on the half-blocks of data bits is invertible. The Data Encryption Algorithm's function f is a cipher since it performs both substitutions and permutations.

The DES algorithm encrypts messages in blocks of 64 bits, which is equivalent to 16 hexadecimal digits. The keys that DES employs to encrypt data are 16 hexadecimal digits long, or 64 bits. The DES algorithm uses a 56-bit key, but discards every eighth bit as noise. In any event, 64-bits (such as 16 Hexadecimal digits) is the round number based on which DES is structured. DES algorithm is based on the fundamental parts: Sub-keys generation and encryption process (Zhou & Li, 2014). The process of encryption using DES cryptographic method consists of the eight-step process (Paar & Pelzl, 2010; Stallings, 2014):

1) Convert plaintext and the key that will be processed into binary bits. Plaintext and the key that has been converted and then broken down into data blocks form with each of the block has a 64 bits (eight bytes) length. If the message is in the form of alphabet or symbols, it must first be converted into decimal and hexadecimal form following the ASCII character table, and then converted into binary bit.

2) Randomize the bits in plaintext data block based on Initial Permutation (IP) table, so that the bit sequence randomized compared to bits sequence of early plaintext block. The bits sequence after the second step followed the results from the IP table, with the first bit derived from the 58^{th} sequence bit of early plaintext blocks, and then the second bit derived from the 50^{th} sequence bits until the 64^{th} sequence bit derived from the seventh sequence bits.

3) The scrambling of key bits based are on permuted Choice 1 (PC-1) table. Results from the PC-1 has a 56 bit length because the last bits of each byte of the key (8, 16, 24, 32, 40, 48, 56 and 64 bits) that acts as the parity bit are not used again in the next step process. Once completed the results of PC-1 then divided into C0 and D0, with C0 is 28 leftmost bits and D0 is 28 rightmost bits from PC-1 results bits sequence.

4) Shift bits to the left (left shift) at Ci and Di as much as one or two times, with the value of i based on encryption process round that consists of 16 rounds. The result of the shift bits from every round of the Ci and Di are then combined into CiDi with a length of 56 bits. After that the CiDi key bits are randomized based on PC-2 (permuted Choice 2) table until produce the Ki variable.

5) Running the data expansion process of Ri-1 with a length of 32 bits (start-

ing from the R0 of second process step results) becomes Ri with a length of 48 bits, where i is the round during the process. This process will be carried out as much as 16 times with the value of turnover $1 \leq i \leq 16$ using the Expansion Table. The results of the expansion process is referred to as E (Ri-1), starting from E (R0) to E (R15). Afterwards, E (Ri-1) will be XOR processed with Ki that has been obtained from the fourth step process for each bit corresponds to running process round to produce Ai variable with a length of 48 bits and in a vector form.

6) Once obtained, Ai then is broken down into eight blocks with each block consisting of six bits. Each block is then distributed into eight pieces of S – Box (Substitution Box), with the first block distributed to the S – Box 1, the second block distributed to the S – Box 2 and so on. The result of the substitution process using S – Box will be collected and produce Bi variable.

7) Once Bi variable is obtained, the next step is to do permutation process on each bit of Bi variable using P – Box table. The results obtained from the permutation using P – Box referred to as P (Bi), with i adapted to the round during the process, starting from P (B1) through P (B16). Thereafter, P (Bi) will be XOR processed with the Li-1obtained from the second process step in accordance with the processes running round to produce a Ri variable with a length of 32 bits and in a vector form. Ri results will then be merged with Li, which came from the Ri-1, into LiRi which is the result of the encryption process of plaintext for each round process with a length of 64 bits.

8) The eighth process step is carried out when the seventh process step has obtained the L16 and R16 from the 16^{th} process round. The next step is the process of reversing positions on L16 and R16, and then combined to obtain the R16L16 form. These results are then permutated using IP-1 (Inverse Initial Permutation) table. Results obtained from the eight process step are referred to as cipher, which is a data block that has been encrypted and is ready to be sent to the recipient along with the other ciphers. A combination of several ciphers is called ciphertext.

The first and second step of the process is done only once at the beginning of the DES encryption process, while the eight step of the process is done only once at the end of the DES encryption process. The third to the seventh process step are carried out 16 times according to the number of rounds of Feistel process used by DES cryptographic method. Even though DES has high encryption ratio, it is open to attacks because of its shorter keys. In addition, it is susceptible to brute force attacks and as such is not suitable to be utilized in M-banking.

4.1.7. Triple-Data Encryption Standard

Triple-Data Encryption Standard (3DES) is the development of DES cryptographic method. The difference between the two methods is 3DES uses triple times the DES process step used in encryption and decryption process by using three key combinations (Rao, 2015). In addition, the effective length of the key used for encryption and decryption process using 3DES cryptographic method is

168 bits (consisting of three sub-key that each have a length of 56 bits), in contrast to DES cryptographic method that uses a key with and effective length of 56 bits (Mathur & Kesarwani, 2013).There are three options to use a combination of sub-key that has become standard in the encryption and decryption process using 3DES cryptographic method (Kumar & Rajaanadan, 2016):

1) Three sub-keys have different combinations (3K3DES).

2) K1 and K2 have different combinations, whereas K1and K3 has the same combinations (2K3DES).

3) Three sub-keys have the same combinations.

From the three options, the use of sub-key, the first option is the best because the three sub-key has a different combination, with an effective key length of 168 bits, so that the data is encrypted using the first option is more difficult to resolve than the use of the second and third options (Kahate, 2003). The second option has an effective key length of 112 bits, because the first sub-key has same combination as the third sub-key, but this option is still better than using the DES encryption process twice. The third option is the weakest compared to the previous option because the first sub-key and the second sub-key negated each other in the process so that the key used in this option has an effective length of 56 bits, the same as the length of the key used by DES cryptographic method. 3DES has moderate encryption ration and its speed is relatively fast. However, 3DES suffers from brute force attacks, chosen plaintext, and known plaintext attacks. Even though it can be used in applications such as smart cards and e-payments, it is not suitable to be utilized in M-banking applications.

4.2. Steganography

Steganography is the process of incorporating a secret message into a cover medium while entirely obfuscating the fact that it exists (Hashim, Rahim, Johi, Taha, Al-Wan, & Sjarif, 2018; Douglas, Bailey, Leeney, & Curran, 2018; Mishra, Yadav, Trivedi, & Shrimali, 2018). According to Morkel (2012), the secret communication can take the shape of plaintext, an image, cipher text, or anything else that can be represented as a bit. Sometimes a stego-key (secret key) that must be known in order to detect and extract the secret message is used as a parameter in the embedding process. A communication is referred to as a stego-object once it has been cloaked in a cover message. The sender must modify the secret message first, and then work with some of the cover object's components to create the stego-object before embedding the information in the cover media (Cao, Wang, Zhao, Zhu, & Xu, 2018). The stego-object is then sent to the appropriate receiver through a communication method. To extract the concealed message, the operation is carried out backwards after it has been received. Prior to transmitting the stego-object, both parties (sender and receiver) must have access to the secret key if the process calls for it (Al-Husainy & Uliyan, 2018).

Steganography is generally used in the communication of secret and when to-

tal freedom is desired. Communication security is very important in both censured and monitored surroundings. Private communications which cannot be secured through cryptography can be secured with steganography (AL-Shaaby, 2017). However, Conklin (Conklin et al., 2015) suggested the use of steganography with other security mechanisms for the provision of layered security as an intruder who succeeds at one layer is still required to bypass the other levels to be completely successful. Communications in the military and intelligence fields require no obstruction; even with content encryption, the detection of a signal can result in an attack on the sender on a modern battlefield (Khanam & Verma, 2018). Such signals can be hidden through steganography. Information that is not intended to be shared with anyone can also be stored using steganography. Other sensitive information such as banking information can also be concealed in a cover object and stored on a private computer (Devadiga, Kothari, Jain, & Sankhe, 2017; Hashim, Rahim, Shafry, & Alwan, 2018).

Different steganographic algorithms have been deployed to ensure data security. It should be noted that not all steganography systems operate with secret keys; however, the security of steganographic systems can be enhanced by applying the Kerckhoff principle. The principle implies that even if an intruder knows the design and implementation of the steganographic system, he must have the secret key to launch a successful attack on the system. Therefore, it may be wise to incorporate the secret keys (public or private) when implementing steganographic systems (Morkel, 2012).

Steganography provides sensitive information security through the embedding of the information in cover media; thus, there is confidentiality. Such hidden information can only be revealed using a steganographic key (AL-Shaaby, 2017). However, the technique and manner used to conceal the information could also serve as identity proofs. The technique for embedding the information, can become a shared secret if wrongly done, can be a mode of identification and authentication (Morkel, 2012). The embedded information cannot be subjected to integrity check because the information may have been altered intentionally or unintentionally, and the changes made to the extracted information may not be observed (Domain, 2018).

Computer scientists and security analysts have recently recognized the security threats posed by the illicit use of steganographic techniques in the global information space (Siper, Farley, & Lombardo, 2005). Terrorists can utilize steganography to communicate secretly without the knowledge of the law enforcement agencies. Owing to this, studies have been on going to find the problems of the existing steganographic systems which can be exploited for hidden information detection, extraction, and/or destruction. There are two major techniques in steganalysis; visual analysis and statistical analysis.

The aim of visual analysis is to reveal the presence of hidden information through a naked eye or computer-aided inspection. Statistical analysis tries to reveal small alterations in the carrier objects (it tries to unravel the statistical

features associated with steganographic processes) (Hussain, Wahab, Idris, Ho et al., 2018). Furthermore, secret information can be removed by email firewall when filtering images and this is another threat to image steganography. However, most of the proposed image steganographic techniques do not rely on e-mail as a communication channel, rather, on websites which can also distribute stego images (Siper, Farley, & Lombardo, 2005).

4.3. Hybrid Algorithms

Hybrid algorithms involve a combination of two or more algorithms to add a level of security for a system. This may involve a combination of two or more steganographic techniques, a combination of two or more cryptographic techniques or a combination of various algorithm techniques. Dhamija and Dhaka (2015) proposed an encryption system that combines embedding methods based on cryptography and steganography schemes. Regarding the cryptographic component, utilization of cryptography and steganography is an attractive mechanism that compliments the security features of the two algorithms. They advised using the frequently used LSB steganography. However, since a single key is used for both encryption and decryption and might be compromised, there is still a problem with key management and control.

Image steganography technique proposed by Pillai, Mounika, Rao, and Sriram (2016) employed DES algorithm to encrypt text communications. The approach uses a block size of 64 bits and a 16 round. Later, the given image was clustered into several segments using the pixel clustering of the k-means algorithms in order to incorporate sensitive data in each segment. Several clustering techniques were employed in the segmentation of the images. A collection of pixel-shaped data was segmented, and as a result, each pixel was broken down to red, green, blue color components. LSB method is then divided into K numbers of tiny segments to be embedded within each cluster after the construction of these clusters. Despite all of these factors, the use of this application is unsafe due to the use of DES and 56-bit key used for encryption. The approach was put out for increasing the stego image's performance capability (Joseph & Sivakumar, 2015). This method promoted the use of AES with the Adaptive Pixel Value Differencing technique for steganography.

A performance analysis survey was carried out using LSB substitution technique in (Padmavathi & Kumari, 2013) on a number of algorithms, including RSA, DES, and AES. The study focuses on the three encryption approaches according to their effectiveness in any application. It also showed that AES is stronger than RSA and DES since it uses less buffer space and decodes and encodes data much more quickly.

A system that incorporates RSA technology and LSB audio steganography to embed encrypted data into audio file in which the message's recipient first separates the encrypted text from the audio before using the RSA decryption algorithm to unlock it was proposed. As a result, the technique enhances the com-

bined properties of the employed steganography and cryptography while providing a greater level of data protection (Gambhir & Mishra, 2015). The system is typically susceptible to factorization and brute force assaults when classic RSA methods are used, making it simple for an attacker to break through.

A study that conceals images using Blowfish cryptographic algorithm was proposed in (Sharma, Mithlesharya, & Goyal, 2013). When assessing the different symmetric algorithms, the adoption of Blowfish algorithm was taken into consideration due to its strength, speed, and great efficiency. On the other hand, a hybrid algorithm that combines ECC with LSB steganography in (Saranya & Thirumal, 2014) provided security services such as availability, mutual authentication, non-repudiation, and data integrity. In comparison to General Packet Radio Service (GPRS), the suggested system's cost per transaction was lower. A study in (Islam, Kobita, Rumi, Karim, & Tabassum, 2021) proposed a method that combines RSA and DSA algorithms in their work. The method generates two keys: the signer's personal key and their public key, enabling the use of public key for decryption in the event that their personal key is used for encryption. To verify authentication, the recipient's mobile phone sends the sender a One Time Password (OTP), which the system then verifies.

A hybrid algorithm proposed by Abdelfattah, Awad, and Nasr (2019) utilized Elliptic Curve Signcryption and certificateless cryptography for M-banking. This system allows sending of documents and multimedia through M-banking applications. Results of the scheme demonstrate that the algorithm performs better than other earlier methods. A summary of AES algorithm provided by Abdullah (2017) was compared to other algorithms like DES, 3DES, and Blowfish. The author lists several of the AES algorithm's salient features and offers findings of past research on it that evaluated how effectively it worked to encrypt data under various circumstances. According to the study, AES is capable of providing security than competing algorithms like DES and 3DES. The best algorithms for M-banking are chosen based on many factors, including security, battery usage, time usage, attack resistance, storage consumption, and compatibility with hardware and software. The most crucial factor is speed, followed by the system's resilience to attackers as illustrated in **Table 6** by (Padmavathi & Kumari, 2013; Mahajan & Sachdeva, 2013; Singhal & Singhal, 2016; Mathur & Kesarwani, 2013; Sengel, Aydin, & Sertbas, 2020).

Table 6 demonstrates how the different encryption algorithms in this study perform. In terms of speed, all algorithms are fast except DSA and RSA. Security attacks indicate that AES is susceptible to chosen plaintext and known plaintext attacks, DES is susceptible to brute force attacks, 3DES is susceptible to brute force, chosen plaintext, and known plaintext attacks, Blowfish is susceptible to dictionary attacks, DSA has no signature verification, RSA is susceptible to timing attacks, and ECC is susceptible to public parameters.

In terms of application areas, AES is recommended for wireless communication and banks, DES for image processing, 3DES for smart cards and e-payments,

Table 6. Analysis of encryption algorithms.

Algorithms	AES	DES	3DES	BLOWFISH	DSA	RSA	ECC
Encryption Ratio	High	High	Moderate	High	High	High	High
Speed	Fast	Fast	Fast	Fast	Slow	Slow	Fast
Key Length	128 - 192 Or 256 bit	56-bit Key	112 - 168 bits	32 bits to 448 bits.	2048 - 3072 bits	1024 - 2048 bits	160
Tunability	No	No	No	Yes	No	Yes	Yes
Security Against Attacks	ChosenPlain Known-Plain text.	Brute Force	Brute Force, Chosen-Plain text, Known Plain text	Dictionary Attacks	signature verification	Timing Attacks	Public limits
Application	Wireless communication, Banks	Image processing	Smart Card, e-payment	Database Security, Ecommerce Software	Web application and email verification	Internet Banking	Web and mobile key exchange

Blowfish for database security and e-commerce software, DSA for web application and e-mail verification, RSA for internet banking, and ECC for key exchange over web and mobile. From statistics in Table 6, AES outperforms the other algorithms in terms of security and speed and is recommended for M-banking.

This paper reviewed different encryption algorithms used for secure data transmission. Most of the reviewed algorithms are utilized in different application areas such as image processing, smart cards, e-payments, database security, e-commerce, web applications, e-mail verification, and key exchange over the web. Only AES algorithm is applied in M-banking because it is a robust algorithm that is fast in encryption and decryption process. Although AES is secure and recommended standard for M-banking, it is not immune to future attacks. Likewise, steganography is a safe communication technique for hiding messages in cover media to avoid detection. However, steganography is susceptible to visual and statistical analysis attacks. This demonstrates that cryptography and steganography individually are inefficient for data transmission over wireless networks. Therefore, this study establishes an existing gap in the algorithms used for securing data transmission in M-banking. This research gap can be filled by utilizing hybrid algorithms that combine cryptography and steganography to produce hybrid algorithms. Hybrid algorithms benefit the security tenets from each algorithm. For example cryptography contributes encryption and steganography provides data hiding such that if one level of security is discovered by an adversary, then it will be difficult to discover the second level of security.

5. Results and Discussion

Table 6 demonstrates that RSA, ECC, and DSA are robust and can be used for secure data transmission. However, they take long encryption and decryption

time and therefore open to several kinds of attacks. For instance, RSA is vulnerable to timing and brute force attacks, ECC is vulnerable to side channel, power, electromagnetic, error message, fault, and timing attacks. On the other hand, DSA lacks secrecy and is vulnerable to forgeries, known-message attacks, chosen-message attacks, and key-only attacks. These factors make RSA, ECC, and DSA encryption algorithms unsuitable for critical applications such as M-banking.

Conversely, Blowfish algorithm is the quickest block cipher which can be used in database security and internet commerce. Findings from this review indicate that when compared to other encryption algorithms, Blowfish is efficient in terms of time and power consumption (Verma, Guha, & Mishra, 2016). However, its weakness is that it cannot provide non-repudiation. In regard to AES encryption algorithm, it is a fast and secure algorithm that has not been broken so far. AES is used in banking systems, government systems, and high security systems to secure mobile or internet banking (Khelifi, Aburrous, Talib, & Shastry, 2013). Although AES has not been broken so far, future attacks including side channel attacks, timing attacks, algebraic attacks, cube analysis attacks and collision attacks.

Results from steganography methods have been reviewed. The different types of steganography include text, image, audio, video, and network steganography. Among these types of steganography, audio and video steganography have limited techniques that can be applied to hide messages in the cover media. There exist several techniques that can be applied to hide messages in images such as spatial domain, transform domain, compressed domain, LSB technique, pixel value differencing, spread spectrum, and randomized embedding technique. While these techniques can be used to hide information in cover media, LSB technique is the most commonly used. However, LSB has limited undetectability and therefore does not stop adversaries from launching attacks. Even though steganography offers security by concealing messages in a cover media, it is susceptible to steganalysis attacks.

The transition to hybrid algorithms represents a new modern paradigm as a result of assaults on the algorithms reviewed. Thus, a combination of cryptography and steganography provides an additional layer of security. For example, a combination of RSA with AES improves security. However, the cryptosystem cannot be used for M-banking since RSA is slow in encryption and decryption time. In addition, RSA is susceptible to various attacks. On the other hand, a combination of RSA and LSB steganography improves data security. And again, RSA is slow in encryption and decryption time. This puts the hybrid algorithm at risk because if an adversary discovered the presence of hidden message in the LSB steganography algorithm, the long decryption-encryption time of RSA will give the adversary time to hack the system.

A combination of encryption algorithm such as AES with steganography techniques such as LSB provide additional layer of security just like other hybrid algorithms. Additionally, AES is fast in encryption and decryption time and has

not been broken as of date. However, since AES is not immune to future attacks, then a combination of security features from AES and LSB steganography makes it more superior in terms speed and applicability. This study therefore finds a combination of AES with LSB steganography techniques commendable for use in M-banking.

6. Conclusion and Recommendation

In this study, a number of cryptographic algorithms have been thoroughly reviewed to identify the optimal strategy for a certain sector of application. Performance of cryptographic algorithms is measured using the following variables: encryption ratio, speed, key-length, tunability, and security against attack. We concluded that AES is appropriate for wireless communication and banks, DES is applied in image processing, 3DES can be used in smart cards, and e-payments, Blowfish is suitable in applications such as database security and e-commerce. On the other hand, DSA is appropriate for web applications and e-mail verification, RSA is suitable for internet banking, and ECC is recommended for key exchange over web and mobile applications.

This paper concluded that cryptographic and steganographic algorithms are known to be independently ineffective in providing protection to information across networks when used separately; thus, a more effective and secure technique can be accomplished by combining cryptography and steganography techniques. The combination of these strategies would ensure data security is strengthened in order to meet the safety and robustness requirements for transmission of data across insecure networks. Hybrid algorithms, which involve fusion of different steganographic and cryptographic algorithms into one approach can be used to fortify data security since the strengths of the combined methods will be used to overcome their weaknesses when used separately. Combining steganography and cryptography can increase security of secret data because data will be encrypted before being embedded into a cover media.

AES algorithm with LSB steganography techniques can be appropriate for utilization in M-banking. This is due to the fact that AES is a quick encryption and decryption technique that is secure and has not yet been compromised. However, given how quickly technology advances, AES might not always be the safest and secure technique. In order to create a tamper-proof hybrid algorithm for securing data transmission in M-banking, security features from LSB steganography and AES can be combined.

This paper therefore recommends utilization of hybrid algorithms such as AES algorithm and LSB steganography techniques for data transmission in M-banking. This is because such a hybrid algorithm incorporates security features from AES algorithm such as encryption of messages and security features from LSB steganography such hiding messages in cover media before transmission. The security tenet of this hybrid algorithm lies on its hardness to break because even if an adversary discovered existence of hidden message, it will be dif-

ficult to extract since the message is encrypted.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

Abdelfattah, R. I., Awad, S., & Nasr, M. E. (2019). Simplified Hybrid Secure Algorithm for Mobile Banking Application. *Journal of Physics: Conference Series, 1447,* Article ID: 012052. https://doi.org/10.1088/1742-6596/1447/1/012052

Abdullah, A. M. (2017). *Advanced Encryption Standard (AES) Algorithm to Encrypt and Decrypt Data, Cryptography and Network Security* (Vol. 16, 1-11). https://www.researchgate.net/publication/317615794_Advanced_Encryption_Standard_AES_Algorithm_to_Encrypt_and_Decrypt_Data

Al-Hazaimeh, O., Alhindawi, N., Hayajneh, S. M., & Almomani, A. (2013). HANON Chaotic Map-Based New Digital Image Encryption Algorithm. *MAGNT Research Report, 2,* 261-266.

Al-Husainy, M. A. F, & Uliyan, D. M. (2018). Image Steganography Technique Based on Extracted Chains from the Secret Key. *Journal of Engineering and Applied Sciences, 13,* 4235-4244.

AL-Shaaby, T. (2017). Cryptography and Steganography: New Approach. *Transactions on Networks and Communications, 5,* 25. https://doi.org/10.14738/tnc.56.3914

Amrita, K. M, Gupta, N., & Mishra, R. (2018). An Overview of Cryptanalysis on AES. *International Journal of Advance Research and Engineering, 7,* 638-646.

Anada, H., Yasuda, T., Kawamoto, J., Weng, J., & Sakurai, K. (2019). RSA Public Key with Inside Structure: Proofs of Key Generation and Identities for Web-of-Trust. *Journal of Information Security, 45,* 10-19. https://doi.org/10.1016/j.jisa.2018.12.006

Anwar, N. B, Hasan, M., Hasan, M., Loren, J. Z, & Hossain, S. M. J. (2019). Comparison Study of Cryptography Algorithms and Its Applications. *International Journal of Computer Networks and Communication Security, 7,* 96-103.

Bhanot, R., & Hans, R. (2015). A Review and Comparative Analysis of Various Encryption Algorithms. *International Journal of Security and its Applications, 9,* 289-306. https://doi.org/10.14257/ijsia.2015.9.4.27

Bhaskar, Ch. U., & Mohan, A. K. (2019). A Novel Way to Encrypting Text and Images Using Elliptic Curve Cryptography. *International Journal of Innovative Technology and Exploring Engineering, 8,* 302-206.

Bunder, M., Nitaj, A., Susilo, W., & Tonien, J. (2017). A Generalized Attack on RSA Type Cryptosystems. *Theories in Computer Science, 704,* 74-81. https://doi.org/10.1016/j.tcs.2017.09.009

Cao, Y., Wang, Y., Zhao, X., Zhu, M., & Xu, Z. (2018). June Cover Block Decoupling for Content Adaptive H.264 Steganography. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security* (pp. 23-30). Association for Computing Machinery. https://doi.org/10.1145/3206004.3206014

Chande, M. K., Lee, C. C., & Li, C. T. (2018). Cryptanalysis and Improvement of a ECDLP Based Proxy Blind Signature Scheme. *Journal of Discrete Mathematical Sciences and Cryptography, 21,* 23-34. https://doi.org/10.1080/09720529.2017.1390845

CommonLounge (2018).

https://www.commonlounge.com/blowfish-the-first-well-known-encryption-algorithm-in-public-domain-770421d468e8416aa3ff0889c15fa214/

Conklin, W. A., White, G., Cothren, C., Davis, R., & Williams, D. (2015). *Principles of Computer Security*. McGraw-Hill Education Group.

Devadiga, N., Kothari, H., Jain, H., & Sankhe, S. (2017). E-Banking Security Using Cryptography, Steganography and Data Mining. *International Journal of Computer Applications, 164,* 26-30. https://doi.org/10.5120/ijca2017913746

Dhamija, A., & Dhaka, V. A. (2015). Novel Cryptographic and Steganographic Approach for Secure Cloud Data Migration. In *Proceedings of the 2015 International Conference on Green Computing and Internet of Things, (ICGCIoT)* (346-351). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICGCIoT.2015.7380486

Domain, W. T. I. S. (2018). A Review and Open Issues of Diverse Text Watermarking Techniques in Spatial Domain. *Journal of Theoretical and Applied Information Technology, 96,* 5819-5840.

Douglas, M., Bailey, K., Leeney, M., & Curran, K. (2018). An Overview of Steganography Techniques Applied to the Protection of Biometric Data. *Multimedia Tools and Applications, 77,* 17333-17373. https://doi.org/10.1007/s11042-017-5308-3

Falade, P. V., & Ogundele, G. B. (2022). Vulnerability Analysis of Digital Banks' Mobile Applications. *NDA Journal of Military Science and Disciplinary Studies, 1,* 44-55.

FIPS 197 (2023). *Advanced Encryption Standard (AES). Computer Security* (pp. 1-36). National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.197-upd1.pdf

Gambhir, A., & Mishra, A. R. (2015). Crypticsteganography: A New Data Hiding Technique with Multilayer Security System. *International Journal of Innovations & Advancement in Computer Science, 4,* 134-136.

Gaur, N. A., Mehra, & Kumar, P. (2018). Enhanced AES Architecture Using Extended Set ALU at 28 nm FPGA. In *5th International Conference on Signal Processing and Integrated Networks* (pp. 347-440). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/SPIN.2018.8474090

Granjal, J., Monteiro, E., & Silva, J. S. (2015). Security for the Internet of Things: A Survey of Existing Protocols and Open Research Issues. *IEEE Communications Surveys & Tutorials, 17,* 1294-1312. https://doi.org/10.1109/COMST.2015.2388550

Hashim, M. M., Rahim, M. S. M., Johi, F. A., Taha, M. S., Al-Wan, A. A., & Sjarif, N. N. A. (2018). An Extensive Analysis and Conduct Comparative Based on Statistical Attach of LSB Substitution and LSB Matching. *International Journal of Engineering & Technology, 7,* 4008-4023.

Hashim, M., Rahim, M., Shafry, M., & Alwan, A. A. (2018). A Review and Open Issues of Multifarious Image Steganography Techniques in Spatial Domain. *Journal of Theoretical & Applied Information Technology, 96,* 956-977.

Hsiao, F. H. (2017). Applying Elliptic Curve Cryptography to a Chaotic Synchronization System: Neural-Network-Based Approach. *International Journal of Systems Science, 48,* 3044-3059. https://doi.org/10.1080/00207721.2017.1364446

Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T., & Jung, K. H. (2018). Image Steganography in Spatial Domain: A Survey. *Signal Processing: Image Communication, 65,* 46-66. https://doi.org/10.1016/j.image.2018.03.012

Islam, A., Kobita, S., Rumi, L. S., Karim, R., & Tabassum, T. (2021). Data Security System for a Bank Based on Two Different Asymmetric Algorithm Cryptography. In V. Suma, N. Bouhmala, & H. X. Wang (Eds.), *Evolutionary Computing and Mobile Sustainable*

Networks (pp. 837-844). Springer. https://doi.org/10.1007/978-981-15-5258-8_77

Jahan, I., Asif, M., & Rozario, L. J. (2015). Improved RSA Cryptosystem Based on the Study of Number Theory and Public Key Cryptosystems. *American Journal of Engineering Research, 4,* 143-149.

Jani, H. B. (2015). Latest Side Channel Attacks and Its Countermeasures Attacks: Attacks Based on Cryptography. *International Journal of Computer Science and Information Technology Research, 3,* 427-441.

Joseph, F., & Sivakumar, S. (2015). Advanced Security Enhancement of Data before Distribution. *International Journal of Engineering Research and General Science, 3,* 1363-1367.

Joshi, K., Gill, S., & Yadav, R. A. (2018). New Method of Image Steganography Using 7th Bit of a Pixel as Indicator by Introducing the Successive Temporary Pixel in the Gray Scale Image. *Journal of Computer Networks and Communications, 2018,* Article ID: 9475142. https://doi.org/10.1155/2018/9475142

Kahate, A. (2003). *Cryptography and Network Security*. Tata McGraw-Hill Publishing Company Limited.

Khanam, M. S., & Verma, M. J. A. (2018). *Novel and Efficient Video Steganography Approach for Enhanced Security*.

Khelifi, A. (2013). Enhancing Protection Techniques of E-Banking Security Services Using Open-Source Cryptographic Algorithms. In *14th International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing* (pp. 89-95). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/SNPD.2013.47

Khelifi, A., Aburrous, M., Talib, M. A., & Shastry, P. V. S. (2013). Enhancing Protection Techniques of E-Banking Security Services Using Open-Source Cryptographic Algorithms. In *14th International Conference on Software Engineering, Artificial Intelligence, Networking and Pararrel/Distributed Computing* (pp. 89-95). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/SNPD.2013.47

Kumar, N., Thakur, J., & Kalia, A. (2011). Performance Analysis of Symmetric Key Cryptography Algorithms: DES, AES and Blowfish. *International Journal of Engineering Sciences, 4,* 28-37.

Kumar, P., & Rajaanadan, N. S. (2016). Data Encryption and Decryption Using by Triple DES Performance Efficiency Analysis of Cryptosystem. *International Journal of Innovative Research in Computer and Communication Engineering, 4,* No. 3.

Kumar, T. M., Reddy, K. S., Rinaldi, S., Parameshachari, B. D., & Arunachalam, K. (2021). A Low Area High Speed FPGA Implementation of AES Architecture for Cryptography Application. *Electronics, 10,* Article No. 2023. https://doi.org/10.3390/electronics10162023

Lin, X., Sun, L., & Qu, H. (2018). An Efficient RSA-Based Certificateless Public Key Encryption Scheme. *Discrete Applied Mathematics, 241,* 39-47. https://doi.org/10.1016/j.dam.2017.02.019

Liu, Z., Huang, X., Hu, Z., Khan, M. K, JeongSeo, H. W. A., & Zhou, L. (2017). On Emerging Family of Elliptic Curves to Secure Internet of Things: ECC Comes of Age. *IEEE Transactions on Dependable and Secure Computing, 14,* 237-248.

Lula, P., Dospinescu, O., Homocianu, D., & Sireteanu, N. A. (2021). An Advanced Analysis of Cloud Computing Concepts Based on the Computer Science Ontology. *Computers, Materials & Continua, 66,* 2425-2443. https://doi.org/10.32604/cmc.2021.013771

Luy, E., Karatas, Z. Y., & Ergin, H. (2016). Comment on "An Enhanced and Secured RSA

Key Generation Scheme". *Journal of Information Security Application, 30,* 1-2. https://doi.org/10.1016/j.jisa.2016.03.006

Mahajan, P., & Sachdeva, A. (2013). A Study of Encryption Algorithms AES, DES and RSA for Security. *Global Journal of Computer Science and Technology, 13,* 15-22.

Mathur, M., & Kesarwani, A. (2013). Comparison between Des, 3des, Rc2, Rc6, Blowfish and AES. In *Proceedings of National Conference on New Horizons in IT-NCNHIT* (Vol. 3, pp. 143-148).

McDonald, N. G. (2020). *Past, Present and Future Methods of Cryptography and Data Encryption: A Research Review*. Master's Thesis, University of Utah.

Meng, X., & Zheng, X. (2015). Cryptanalysis of RSA with Small Parameter Revisited. *Information Process Letters, 115,* 858-862. https://doi.org/10.1016/j.ipl.2015.06.013

Mishra, S., Yadav, V. K., Trivedi, M. C., & Shrimali, T. (2018). Audio Steganography Techniques: A Survey. In S. K. Bhatia, K. K. Mishra, S. Tiwari, & V. K. Singh (Eds.), *Advances in Computer and Computational Sciences* (pp. 581-589). Springer. https://doi.org/10.1007/978-981-10-3773-3_56

Mitra, S., Jana, B., Bhattacharya, S., Pal, P., & Poray, J. (2017). Quantum Cryptography: Overview, Security Issues and Future Challenges. In *4th International Conference on Opto-Electronics and Applied Optics* (pp. 1-7). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/OPTRONIX.2017.8350006

Morkel, T. (2012). *Image Steganography Applications for Secure Communication*. Doctoral Dissertation, University of Pretoria.

Nawaz, M., Motiwalla, L., & Deokar, A. V. (2018). Usage-Driven Personalised Mobile Banking Application: A Research Prototype. In *The Proceedings of the 2018 ACM SIGMIS Conference on Computers and People Research* (p. 159). Association for Computing Machinery. https://doi.org/10.1145/3209626.3209736

Nie, T., & Zhang, T. (2009). A Study of DES and Blowfish Encryption Algorithm. In *TENCON 2009 IEEE Region 10 Conference* (pp. 1-4). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/TENCON.2009.5396115

Paar, C., & Pelzl, J. (2010). *Understanding Cryptography: A Textbook for Students and Practitioners*. Springer. https://doi.org/10.1007/978-3-642-04101-3

Padmavathi, B., & Kumari, S. R. (2013). A Survey on Performance Analysis of DES, AES and RSA Algorithm along with LSB Substitution Technique. *International Journal of Science and Research (IJSR), 2,* 170-174.

Pillai, B., Mounika, M., Rao, P. J., & Sriram, P. (2016). Image Steganography Method Using K-Means Clustering and Encryption Techniques. In *International Conference on Advances in Computing, Communications and Informatics* (pp. 1206-1211). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICACCI.2016.7732209

Purohit, S., & Arora, R. (2021). Adoption of Mobile Banking at the Bottom of the Pyramid: An Emerging Market Perspective. *International Journal of Emerging Market, 18,* 200-222. https://doi.org/10.1108/IJOEM-07-2020-0821

Rao, S. (2015). Performance Analysis of DES and Triple DES. *International Journal of Computer Applications, 130,* 30-34. https://doi.org/10.5120/ijca2015907190

Saini, A., & Vandana, D. A. (2022). Study on Modified RSA Algorithm in Network Security. *International Research Journal of Modernization in Engineering Technology Science, 4,* 1461-1465.

Sakala, L., & Phiri, J. (2019). Factors Affecting Adoption and Use of Mobile Banking Services in Zambia Based on TAM Model. *Open Journal of Business and Management, 7,*

1380-1394. https://doi.org/10.4236/ojbm.2019.73095

Saranya, J., & Thirumal, S. (2014). Framework for Secure Mobile Banking Application Using Elliptic Curve Cryptography and Image Steganography. *International Journal of Scientific Engineering and Research, 2,* 48-50.

Sari, W. S., Rachmawanto, E. H., & Sari, C. A. (2017). A Good Performance OTP Encryption Image Based on DCT-DWT Steganography. *Telkomnika, 15,* 1987-1995. https://doi.org/10.12928/telkomnika.v15i4.5883

Sengel, O., Aydin, M. A., & Sertbas, A. (2020). Determining the Cryptography Algorithm and Model for Mobile Payment Services. *Acta Infologica, 4,* 21-33.

Seo, J. H. (2020). Efficient Digital Signatures from RSA without Random Oracles. *Information Science, 512,* 471-480. https://doi.org/10.1016/j.ins.2019.09.084

Sethi, D., & Acharya, D. (2018). Financial Inclusion and Economic Growth Linkage; Some Cross-Country Evidence. *Journal of Financial Economic Policy, 10,* 369-385. https://doi.org/10.1108/JFEP-11-2016-0073

Sharma, K., Agrawal, A., Pandey, D., Khan, R. A., & Dinkar, S. K. (2022). RSA Based Encryption Approach for Preserving Confidentiality of Big Data. *Journal of King Saudi University-Computer and Information Sciences, 34,* 2088-2097. https://doi.org/10.1016/j.jksuci.2019.10.006

Sharma, M. H., Mithlesharya, M., & Goyal, D. (2013). Secure Image Hiding Algorithm using Cryptography and Steganography. *Journal of Computer Engineering, 13,* 1-6. https://doi.org/10.9790/0661-1350106

Sharma, N., & Bohra, B. (2017). Enhancing Online Banking Authentication Using Hybrid Cryptographic Method. In *Proceedings of the 3rd International Conference on Computational Intelligence and Communication Technology* (pp. 1-8). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/CIACT.2017.7977275

Simplilearn (2022). *Digital Signature Algorithm (DSA) in Cryptography: How It Works and Advantages.* https://www.simplilearn.com/tutorials/cryptography-tutorial/digital-signature-algorithm

Singhal, S., & Singhal, N. (2016). A Comparative Analysis of AES and RSA Algorithms. *International Journal of Scientific & Engineering Research, 7,* 149-151.

Siper, A. P., Farley, R., & Lombardo, C. (2005). *The Rise of Steganography.* https://api.semanticscholar.org/corpusID:110715828

Stallings, W. (2014). *Cryptography and Network Security-Principles and Practice* (6th ed.). Pearson.

Stinson, D. R. (2006). *Cryptography, Theory and Practice* (3rd ed.). Chapman & Hall/CRC. https://doi.org/10.1201/9781420057133

Tahir, A. S. (2015). Design and Implementation of RSA Algorithm Using FPGA. *International Journal of Computer and Technology, 14,* 6361-6367. https://doi.org/10.24297/ijct.v14i12.1737

Thapar, S. S., & Sarangal, H. (2018). A Study of Data Threats and the Role of Cryptography Algorithms. In *IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference* (pp. 819-824). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/IEMCON.2018.8614943

Thiyagarajan, R., & Meenakshi, P. B. (2019). An Enhancement of EAACK Using P2P ACK and RSA Public Key Cryptography Meas. *International Journal Measurement Confederation, 136,* 116-121. https://doi.org/10.1016/j.measurement.2018.12.031

Valmik, N. K., & Kshirsagar, V. K. (2014). Blowfish Algorithm. *Journal of Computer En-*

gineering, 16, 80-83. https://doi.org/10.9790/0661-162108083

Verma, A., Guha, P., & Mishra, S. (2016). Comparative Study of Different Cryptographic Algorithms. *International Journal of Emerging Trends & Technology in Computer Science, 5,* 58-63.

Zhou, Y. B., & Li, Y. Z. (2014). The Design and Implementation of a Symmetric Encryption Algorithm Based on DES. In *5th International Conference on Software Engineering and Service Science* (pp. 517-520). The Institute of Electrical and Electronics Engineers. https://doi.org/10.1109/ICSESS.2014.6933619