Article

An Improved SMS Security Technique to Avoid Plaintext and Dictionary Attacks

Humaira Ashraf [*] , Mirza Qurat Ul Ain Zam Zam , NZ Jhanjhi [*]

Posted Date: 8 January 2024

doi: 10.20944/preprints202401.0495.v1

Keywords: Android; Android security; cryptographic techniques; SMS security; Deoxyribonucleic acid (DNA); ECSS

Article

An Improved SMS Security Technique to Avoid Plaintext and Dictionary Attacks

Mirza Qurat Ul Ain Zam Zam [1], Humaira Ashraf [1] and N.Z Jhanjhi [2,*]

[1] Department of computer and software Engineering International Islamic University, Islamabad, P.O Box 44000, Pakistan; mirza.mscs1087@iiu.edu.pk, Humaira.ashraf@iiu.edu.pk

[2] School of Computer Science, SCS Taylors University, Subang Jaya, Selangor, Malaysia

* Correspondence: noorzaman.jhanjhi@taylors.edu.my)

Abstract: Digital communication is becoming an integral component of contemporary society and the use of mobile devices is increasing faster than ever. The function of message communication is becoming more popular as for mobile phone users, a short messaging service (SMS) requires an extremely high level of security. Any SMS that enters a person's smartphone must be considered a private affair, with privacy and security safeguarded. This research study found a novel Deoxyribonucleic acid (DNA) Cryptography method that uses a dynamic DNA sequence table to increase security. Cryptography is a popular approach for ensuring the confidentiality of a message. This paper proposed an Efficient Cryptographic Scheme for SMS (ECSS) algorithm to protect texts received from an Android smartphone. ECSS-based encryption has been identified as a novel method of securing information in the form of DNA molecules that employ DNA strands to conceal the information. This Paper provides the DNA cryptography approach for encrypting and decrypting plain texts in this work. The primary goal of ECSS cryptography is to guarantee secrecy when people transfer data across a network. This paper examines DNA Crypto, the distinction between classical cryptography and DNA Data encryption. An Android-based app is created to perform the proposed technique and the results show that the algorithms produce a consistent encryption result in which the length of the message before encryption is always the same as the length of the message after decryption. The suggested security solution has the potential to greatly improve the security of mobile communication. This Paper also provides experimental results for our chat application performance, such as received text message correctness, cipher text, average encryption time and average decryption time. According to the findings/results of this study, the speed of average encryption and decryption of the description is roughly 0.011 and 0.037 sec and prevents plaintext and dictionary attacks.

Keywords: Android; Android security; cryptographic techniques; SMS security; Deoxyribonucleic acid (DNA); ECSS

1. Introduction

Communication technology is becoming a crucial aspect of modern growth and the use of mobile devices is increasing faster than ever. The function of message communication is becoming more popular as the number of smartphone customers rises. Technology has facilitated a wide range of activities, with most of us relying on it for communication. Globally, there are 5.20 billion smartphone users. Furthermore, Pakistan had 164.9 million mobile influencers and 76.38 million active internet users in 2020. Between 2019 and 2020, the number of internet users in Pakistan increased by 11 million (+17%), although mobile phones increased by a million (approximately 6.2%) [1]. The number of smartphone connections in Pakistan in 2020 was comparable to 75% of the total population, with 68 percent of Smartphones in use. Cellular users in Pakistan touched 183 million at the end of March 2021, up from 178.97 million in January. In Pakistan, there were 85 percent more smartphone connections in 2021 than there were people overall. With over 2.5 billion active users in

190 countries, Android is the most popular operating system in the world. In 2022, the number of smartphone connections in Pakistan was equivalent to 93.33 percent of the entire population [2].

Today's fastest and most convenient mode of communication is messaging. Every year, almost 8 trillion communications are transmitted throughout the world. All cell phones and providers enable short messaging. SMS is sent through the transmission medium; thus no internet connection is required. But SMS had limitations; it was limited to 160 characters and did not make efficient use of emojis. Text messages are prioritized for all phones. Every mobile phone user understands how to read and send text messages. Meanwhile, some businesses are embracing SMS, while others are refusing to do so since SMS is hazardous, with texts easily stolen and retrieved for harmful purposes. Text messaging also falls short when compared to other interactive experiences on cell phones since it is considered archaic [3]. In today's environment, it is necessary to be able to stimulate all human occurrences to boost proficiency, effectiveness and security for consumers. SMS is still frequently used due to its simplicity. Furthermore, a few instances of human negligence, such as misplacing the physical key, have led to the invention of electronic access control, in which only special persons or those who are permitted to reach a site use a technique that only specific individuals are aware of. Others' technique is based on the usage of code or a secret word [4].

In the subject of information knowledge, information security is still a work in progress. Message Security and confidentiality issues are continuously being researched in the information technology era since there is no best security and most efficient solution accessible. For smartphone users, a short messaging service is a typical mode of communication. SMS (Short Message Service) requires an extremely high level of security [5]. Any SMS that enters a person's smartphone must be considered a private affair, with privacy and security safeguarded. Security is an increasingly crucial aspect of the communication medium. Because of its simplicity, SMS is still widely utilized [6]. However, such methods are readily hacked, making SMS a riskier means of transmitting confidential information. Cryptography is a popular approach for ensuring the confidentiality of a message. The problems were uncovered when reviewing the literature. However, such methods are readily hacked, making SMS a riskier means of transmitting confidential information. As a result, correct cryptographic methods must be utilized to preserve security [7]. MS-based phone services need a high level of network security as well as a quick response time [8]. As a result, the focus of this research will be on employing a cryptographic technique to secure SMS. Cryptographic approaches for short message service security are time-consuming and subject to plaintext and dictionary attacks [9].

This scheme paper, purposes a DNA cryptographic algorithm for the security of short messaging services that need less time and are vulnerable to plaintext & dictionary attacks while using a secured cipher. A security technique is presented to keep the text that avoids different plaintext assaults and dictionary attacks. The purpose of information security places a premium on emerging methods of data protection. DNA-based encryption has been identified as a novel method of securing information in the form of DNA molecules that employ DNA strands to conceal the information. The primary goal of DNA cryptography is to guarantee secrecy when people transfer data across a network. This scheme paper examines DNA Crypto. Traditional cryptography approaches are utilized in the current literature. This scheme paper provides SMS protection strategies used in mobile communication systems to increase message security. Furthermore, our scheme paper looked at the open research gaps for Android SMS security.

1. Unauthorized Access: Use better pseudo-random generation for a key stream like elliptic curve implementation of DSA because rc4 does not provide authentication [6].
2. Impersonation Attack and Replay Attack: The protocol has a few flaws, including a failure to key compromise impersonation, an unreliable password-changing process, imperfect shared verification and an internal attack drawback [10].
3. Low Response Time: Rc4 and affine cipher do not work on response time, they only work on the security of encryption and decryption [6]. The approach does not apply to the text if the main length is too long in text and the keys are handed over to a third party, which is a major flaw in symmetric encryption [3]. However, need to optimize the encryption and decryption time, as well as the hyperparameter, to boost our performance [11]. The security has been greatly

improved. As a result, this paper provided this comprehensive literature review to contribute to the work by filling in the gaps in existing surveys.

4. Dictionary Attack: A dictionary attack is a type of assault for which the encoded phrases are very similar to terms found in a dictionary, making it easy for hackers to break the plain text. This scheme paper shows the following are the primary contributions of this study.

1.1. Contributions

1. The encryption and decryption time of the message has been described in this scheme paper.
2. Extract different features (Time, Security) and discover networking methods.
3. Prevents dictionary, plaintext and other passive attacks.
4. Identified current gaps

The remainder of the scheme paper is organized as follows. Section I introduces SMS security, Section II discusses the Literature Review, Section III focuses on the proposed methodology, Section IV gives Mathematical modeling, Section V focuses on the Simulation part, Section VI concludes the results and discussion, Section VII is the conclusion.

2. Literature Review

This section delves further into the research for all of the articles that were chosen. This section discusses several cryptography approaches. The fundamental purpose of this research is to determine the best solution to the issue statement (optimal Algorithm). A combination of two or three synchronous cryptographic algorithms is used to protect the Short messaging service on Android. Android is a free operating system that gives a platform and infrastructure for no identical diggings inventors to create novel activities with distant programming interfaces akin to operational scripting languages. Android offers a comprehensive framework for mobile phone operators, inventors and handset manufacturers all around the world to create breakthrough hardware and software services. Technology companies that are looking for a pre-made, an easily customizable operating system for cutting-edge devices frequently turn to Android. It is an adaptable, user-friendly operating system. When using their phones frequently, mobile phone users want more secure and private communication. The encoding mechanism is a cipher component that uses the logical bitwise operator to change each character of the text with each byte of the resulting keystream. This procedure is quick and easy. The RC4 cipher and the Affine cipher are two forms of stream cipher encryption used to protect SMS. The average encryption time in this paper is less than 0.24 seconds [9].The investigation concluded with an average encryption time of 678.68ms for the text of 100Char and a decryption time of 732.18ms [11]. Advanced Encryption Standard (AES) is a block encoding algorithm with typical key sizes of 128,192 and 256 bits. The state, block, byte, word(char) and bit data sizes are employed in this Algorithm. The user generates the key, RC4 is used to encode the SMS and RC4 is the most recent method. The encryption function was carried out character by character. The encryption and decryption processes are rapid and efficient [12].

In addition to a SIM insertion slot, mainframes, smartphones and tablets now have a variety of functionalities [13]. Several cryptography methods were thoroughly compared. The internet is frequently utilized in current culture for information sharing, commercial decisions and other things. However, using cell services to send and receive quick messages is not a very secure method. To fix the issue, use several methods that address security issues, such as cryptography techniques (DNA, Rivest-Shamir-Adleman (RSA) , ECC and AES Algorithms extreme for data systems). AES is a symmetric encoding method created to take the role of Data Encryption Standard (DES). AES is a block code with key sizes of 128, 192 and 256[14] and a block length of 128 pieces. The round change is made by AES using byte replacement, changing lines, mixing columns and adding round key advances. AES has been broken by both brute-force attacks and other arithmetic methods [15].

For every sender and receiver to efficiently communicate secret information, secure communication is essential. Android smartphones are becoming among the most popular devices so, communications must be conducted in a very secure environment. Numerous methods and systems for encoding and decoding plain text have been developed in the field of computational cryptography to meet these security requirements. The methods and techniques of DNA cryptography, however, defeat these strategies. An important aspect of next-generation security is DNA cryptography. The encoding table should be generated fresh at regular intervals or for each interaction session between the sender and recipient to provide a higher level of security. It is crucial to provide various DNA sequences for each character set component. None of the encoding table creation techniques that are currently available can accomplish this goal. The DNA encryption of the plaintext shall offer a robust encoding strategy that is exceedingly challenging to decode to ensure assault resistance. The biological processes that are simulated to adapt to the digital computer environment should serve as the basis for the DNA encryption and decryption technique. DNA cryptography is currently a developing subject of study [16]. The solution proposed takes into account a biological modeling methodology based on a unique DNA encoding table for each character set. It produces a random encoding table following each sender-receiver interaction session. The DNA algorithm was employed in research to convey encrypted messages. The researchers proposed novel techniques for concealing data based on DNA for optimum safety and strong security with great capacity. DNA-based data-concealing methods have been the subject of a recent study. The majority of them make use of biological features of DNA sequences. It is made up of the most complicated organic compounds. The genetic data is stored in DNA as a sequence of four chemical bases: Adenine (A), Guanine (G), Cytosine (C) and Thymine (T). DNA coding method is based on a symmetric key which means that key groupings are achieved from the hereditary repository and cleared out as they are on both endpoints: sender and beneficiary(receiver). Plain content is firstly changed over to binary organize and after that to DNA organize utilizing the DNA substitution. Asymmetric algorithms using public and private keys are introduced in [17].

Cryptography is a technique for encrypting and decrypting data and it contains several algorithms created by researchers to improve the system's security [18]. Several academics are now experimenting with the Elliptical curve cryptography (ECC) method, which is a swinging structure in encryption (technique). Ramdhansya uses the ECC algorithm to protect text messages in a smartphone messaging app. The key size and security level of ECC are the aspects that drive scholars to investigate and discover it to find its limit and strong point. When compared to other cryptographic methods such as Diffie Hellman and RSA, ECC's key size is comparatively modest and also suitable for mobile devices; it emphasizes a handful of public and private keys for decryption, which entails reconversion of ASCII to Plain message and encryption [19]. For online distribution, convert regular text to ASCII figures [20].

In [21] 4*4 DNA encryption technologies were developed to alter matrices utilizing the primary information of the signal, resulting in exceptionally secure data. Aside from elements that give a robust layer of defense, limitations entail big encrypted information and protection that is solely dependent on the key. The method presented in [22] is based on the notion of a dynamic DNA sequence table, which allocates randomized ASCII characters to the DNA sequence at the start. Then it uses a specific number of copies to alter the ASCII position in the sequence table programmatically depending on a logical term. The use of the one-time pad (OTP) board, on the other hand, makes the technology more efficient because the regular OTP plaintext, as well as the key, should be comparable in size, making secure key transmission harder difficult [23]. The proposed technique performs great with original data. Furthermore, the encrypted procedure employs several co-protections by establishing a dynamic coded table, information dependence and numerous dynamic round keys. [24] presented an asymmetric cryptography approach is divided into five stages:1st generate DNA public and private keys,2nd is the development of a dynamic DNA sequence table,3rd making 14 round keys,4th is the encryption procedure and 5th is a decryption procedure

The Affine encryption algorithm is a Symmetric cryptography algorithm. The Affine code is a delegated monoalphabetic replacement cipher that extends the Caesar cryptograph by increasing the plaintext for certain numbers and adding them with move activity[25]. Because of its straightforward cryptographic calculation, Affine code is better to combine with another calculation to provide incredible encryption to thwart the unscrambling effort of an unauthorized party [26]. To obfuscate non-governmental party decryption efforts, Affine Cipher employs extra approaches for flexible multi-level coding. This research used a well-prepared approach. RSA has teamed up with the Vigner cipher to deliver better outcomes in less time. This combination produces excellent results, but it takes longer to compress the data. The presented research in an article [27] is a modification of the Vigenere cipher by including an asymmetric mechanism into the Vigenere cipher. Because the Vigenere cipher only offers the symmetric technique, which utilizes the same key for both encryption and decryption, [28]suggested an asymmetric approach in which they employ a public key and a private key. In this case, the public key is the inverse of the private key. Encoding the information message from the source is the first stage, followed by coding the key. The next recipient side receives a new encryption key and encrypted text. The findings show that both sets of criteria produce reliable encryption of the results, with the text sizes being the same before and after the encryption-decryption procedure.

The message is encrypted using powerful cryptographic techniques. It employs the RSA algorithm to protect keys and the AES technique to protect messages. The pattern lock is used to create authentication. This technology ensures a safe and dependable information network. Research [29] proposed in-app encrypted texts using the RC4 algorithm as the secure method of delivering text. RC4 does, though, incorporate the Key Scheduling Algorithm (KSA) as well as the Pseudo-Random Generation Algorithm (PRGA). As a result, the RC4 algorithm was often upgraded with a random beginning state to boost the unpredictability of the main channel. The suggested method's performance is measured using encoding and deciphering times and based on the correlation values. Depending on the information, it appears that all amounts of the transmitted SMS words influence the cryptographic operations times, with a maximum correlation of 0.00482[29]. In [30] the survey is conducted on RC4 to enhance the RC4 and remove the weakness of RC4. The suggested algorithm in [31] is based upon the value used in ASCII to encode plaintext. The method used to produce a key for the individual at random that is the same length as the plaintext. The randomly generated key is altered to another key by replacing the position of the key with a random integer and is used to decrypt the original plaintext. In [32], researcher present a symmetric DNA binary encryption algorithm for encrypting and decrypting plaintext data. In this research, the authors [33] introduced a method that runs on the Android platform and encrypts texts before they are sent over the network by the user allowing them to encrypt communications before transmitting data over the network. In the continuum of fortifying SMS security measures, this study builds upon earlier works [34-48] to address the persistent threats posed by plaintext and dictionary attacks.

2.1 Problem Statement

The issues were discovered during the review of the literature. Security is a more important part of the communication medium. Because of its simplicity, the short messaging service (SMS) [6] is still widely utilized. Such systems, however, are easily hacked, making SMS a riskier method of sending private information. As a result, to maintain data security, proper cryptographic methods must be used. MS-based phone services necessarily require a high degree of network security with a fast response time.

This research will hence focus on using a cryptographic algorithm to secure the SMS. Cryptographic methods for the security of short messaging services are time-consuming and vulnerable to plaintext and dictionary attacks [10].

2.2 Solution

In this paper, this research paper suggested a DNA cryptographic algorithm for the security of short messaging services that need less time and are vulnerable to plaintext & dictionary attacks while using a secured cipher. It is robust against other attacks like Man in the Middle (MITM) and other passive attacks. To keep text, a security technique is presented that avoids different plaintext assaults and dictionary attacks.

3. Efficient Cryptographic Scheme for SMS (ECSS)

This section discusses the proposed scheme. The whole methodological configuration is presented. The sender and receiver processes are utilized. To encrypt the communication from the sender's cell phone, the DNA Secure Encryption [31] Algorithm is employed. This encryption method makes use of a random key generation process to generate ciphertext with the help of DNA encryption [32]. The encrypted data will then be transferred to the receiver. The application is likewise present on the receiver side. The suggested strategy is broken into two stages. Each step is intended to take as little computational and reaction time as possible.

3.1 Proposed Scheme

The proposed methodology is separated into two stages. Each step is intended to take as little computational and reaction time as possible. The stages that have been designed are listed below.

Phase 1. Sender and Receiver side
Phase 2. Efficient Cryptographic Scheme for SMS (ECSS)

3.2 Encryption and Decryption Procedure

Encryption and decryption [31] are the two main procedures in our proposed methodology. The ECSS method produces user security based on DNA. The data in the DNA computer is represented by a four-nucleotide sequence: "A," "C," "G," and "T." For the creation of a DNA pattern, a novel suggested encryption approach is applied. The data is encrypted and sent to the next level as input. Then, on the following level, a random number, such as Pn, is produced and utilized for encryption. The decryption procedure is then applied. The entry data is a collection of characters in plain text. Then, each element is treated as a character and converted to the appropriate ASCII code. A binary representation of the ASCII character is created. In Figure 1, the full process of DNA encoding is described. An input message is examined and transferred to the receiver after encryption.

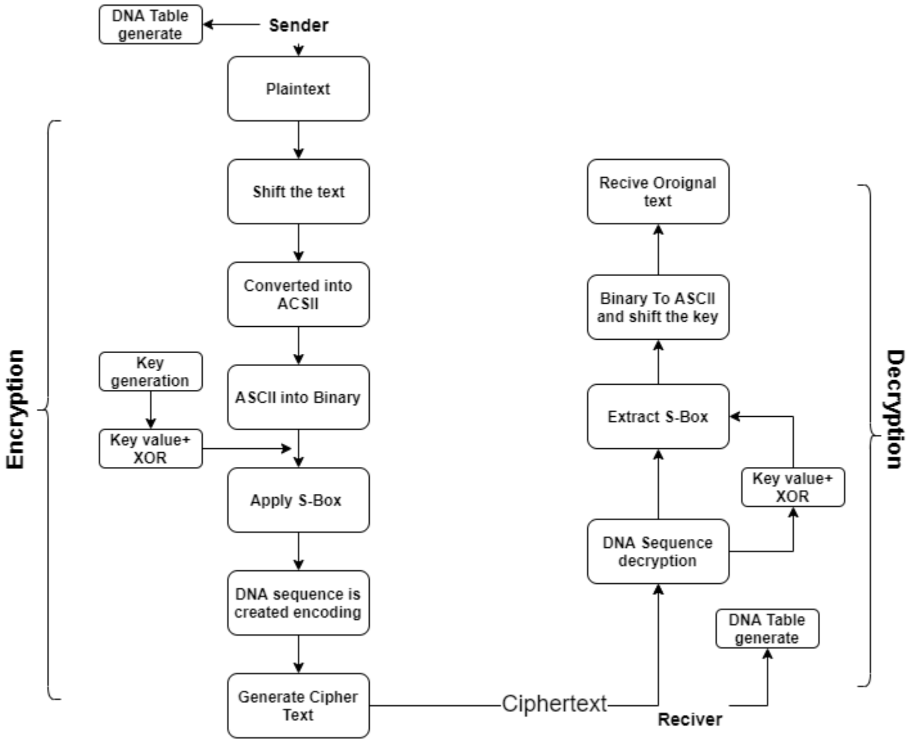

Figure 1. Proposed Scheme.

3.3 Encryption Process

Encryption is the technique of encrypting a communication so that only the authorized person may read it. The produced plaintext from mobile SMS is sent to the server through the transmission media. ASCII is created from plaintext. The plaintext is then transformed into an 8-bit binary using a binary converter. Step 2's 8-bit binary values are then Exclusively-OR (XOR) with the same bit of key produced. Next, split the resultant 8-bit binaries in half. In DS1 the resultant data are stored. It first ignores the first half of the data and only considers the second half. Then, by inserting DNA values, you may turn four bits into eight. Then, on the 8-bit produced data, do shift one. Substitute appropriate bits from the Table 1 DNA table to convert binary data back to 4 bits. Add previously left-over bits from the first half to create 8-bit binary data. Finally, insert 8-bit binary data into the DNA sequence shown in DNA Table 2. The message's ciphertext is encoded in DNA.

Step 1: Original plain text is sifted and converted into ASCII form
Step 2: All ASCII value is converted into binary form (0's and 1's)
Step 3: Random key is generated
Step 4: DNA-based table is applied from Table 1
Step 5: Text is changed into DNA code
Step 6: The message is converted into cipher text

3.4 Algorithm

```
Algorithm: The Proposed ECC algorithm
Input: Plain Text (P)
Output: Cipher Text (C)
Encryption Process
1.    Convert Each character into ASCII
2.    Convert each ASCII into Binary(B)
3.    DS1←encoded DNA sequence of B
4.    Split DS1 into two parts p1 and p2
5.    Now take key value k that is generated by random key generation algorithm and apply XOr
operation on k.
6.    If (P is received) then
7.        Apply S-Box on Binary and DNA sequence is created and binary is converted into C.
8.    Else
9.        if (not received) then
10.       Go to step 5
11.       Else
12.           if (successfully created C) then
13.           C and k are sent to the Receiver side.
14.           End if
15.       End if
16.   End if
17.   End
18.   Get Cipher Text
```

The decryption process is the reverse of the encryption process.

3.5. Decryption

Decoding is the opposite of encryption, in which the ciphertext is turned back into plaintext using any range of steps. The recipient is given encrypted data that is unreadable. First, replace the ciphertext's DNA sequence with its matching binary from S-box. DNA table should be applied to the produced values. Divide the 8-bit information into two parts, each with four bits. The second half of the bits should be ignored since they are DNA-extended bits. Now, use the random key creation to do an XOR operation. After that, perform shift one on 8-bit generated data Afterward all of the decryption processes have been completed and the value is transformed to ASCII decimals and finally to plaintext (original).

Because of its simplicity, ECSS has a shorter computational and reaction time and makes it difficult for attackers to recognize plaintext.

4. Mathematical Modeling

$$P=ID+ Timestamp+ D \tag{1}$$

P is the plain text that is encrypted for security purposes. Based on the Id and timestamp data, P calculates the message format. ID is the message ID that is added with a timestamp value. Send a brief note to the receiver.

$$P' = pA \mid dA \tag{2}$$

P' is the outcome option for the identified assault, which can be either a plaintext attack or a dictionary attack. P' just detects any assault and begins the encryption process. pA is Plaintext Attack and dA is Dictionary Attack.

$$E = P' \rightarrow An \tag{3}$$

$$E' = E.Cn \rightarrow Bi \tag{4}$$

E is the preliminary data that is updated based on the techniques used. In this case, P' transforming randomly selected to input into their matching ASCII numeric form (An). The binary input (Bi) data transformed from coded no(Cn) that is stored in E'.

$$E1 = Bi \oplus K \tag{5}$$

Binary input information is XORed with the key value (K) obtained by key generation and the resulting data is maintained in E1.

$$E2=E1+DSb \tag{6}$$

$$E3=seq+ciptx \tag{7}$$

DNA S-box is applied to the E1 data from Table 1 and Table 2. The resulting data is updated in E2. E3 updates data by performing DNA conversion on E2's created text. It generated encoding DNA sequences. The cipher text stored in the E3 is this sequence-created string.

Table 1. DNA S-Box.

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0	acgt	accg	ttac	acag	ttat	ggag	tatt	atgg	tcga	agta	ccgt	ggaa	gagc	ggtt	aaac	ctgc
1	acac	actg	gcag	gtgt	aacc	agtt	gagt	aggt	accc	cacc	taac	gaac	acca	ccca	caag	ctct
2	aaag	taag	agct	ctaa	tcat	gagg	caac	aggg	tgcc	caca	ccag	ctat	aatg	cgtg	tggt	tcgc
3	atac	cgac	ggta	agtc	gacg	tttg	cgct	tctt	aaat	cgtt	ccct	tggg	gtca	ccaa	gtcc	gaag
4	gtat	gtaa	cctg	gtgg	atcg	atcc	tatc	gggg	ttta	tgcg	caga	tgtc	atat	tcac	tgtt	aacg

5	cctc	aaga	cggg	caat	cttg	atgc	tagg	cgtc	gatt	aaca	ggcc	cttt	gttg	atct	ccga	tctc
6	tgca	ctcg	aatt	taga	tgag	ccta	cagt	gggc	taca	ttgt	ggca	atga	tgta	cctt	aact	atgt
7	gcca	tgga	agac	ttca	ataa	gtct	tagt	catg	cgcc	cgaa	tatg	gaat	ttaa	actc	attg	tata
8	ctta	taaa	gcga	ccat	ggcg	tacg	tacc	attc	cgat	agtg	agat	gcgc	cggt	tgaa	agga	atca
9	aagc	tggc	gacc	gata	tttc	tccc	gtag	aaaa	ttga	cgta	ctgg	agcc	ctcc	attt	gtgc	tgac
A	aagg	gccg	gggt	tcca	tctg	gaaa	ggat	acta	ccgc	gcgg	ctag	tact	agag	ggac	cgcg	ctgt
B	cata	gctg	cact	catt	aatc	ccgg	ctca	ttct	tcaa	gccc	aggc	ctga	cgca	tgtg	cccc	taat
C	acgc	actt	tagc	agcg	catc	agaa	ggtg	agca	gcgt	aata	gttt	ggct	gact	tcta	cggc	ccac
D	aagt	ttcc	ctac	acct	tcgg	acat	gtta	tcag	cagc	ggga	tccg	tcgt	ttgg	gatc	gtcg	cagc
E	tgct	acga	gctt	caaa	tgat	ttag	gaga	ggtc	gcct	cgag	gcaa	cagg	atag	tcct	gcat	gtga
F	gtac	cccg	cgga	tttt	ttcg	atta	gcac	gatg	acaa	gttc	gaca	ttgc	cttc	gctc	acgg	gcta

Table 2. DNA based table.

! →accg	< →gtca	{→tgcc	b→ggca	s→catg	9→ctga	Q→gcga
” →ttac	=→aacc	|→caca	c→ttgg	t→caag	A→gctt	R→aatc
#→acag	>→accc	}→tacg	d→gcgt	u→caat	B→atag	S→gggg
$→ttat	? →cacc	~→ccag	e→cgag	v→gcat	C→gttg	T→gaag
%→atgg	@→gaac	-→ttgt	f→agag	w→cgta	D→gccc	U→ggtc
&→agta	[→acca	. →ctgc	g→acat	x→tcga	E→taac	V→ggtg
’ →ccgt	space→ctaa	/→gcta	h→tccc	y→agct	F→aaag	W→cgtt
(→ggaa	\→ctct	: →aaac	i→aaaa	z→tgac	G→cccc	X→acgt
aa) →gtac	] →agtt	n→tagg	j→cagt	0→aagt	H→tcac	Y→tgca
*→gagc	^→gagt	o→cgaa	k→atgc	1→tgct	I→tagc	Z→actg
+→acac	_→tcat	p→tatg	l→atcg	2→ggtt	J→agat	; →gtgt
, →gcag	`→gagg	q→gaat	m→agtc	3→gatc	K→atac	a→tggg
7→cgta	N→ctca	6→cgat	8→ctag	4→gact	L→gccg	r→tgta
P→ggct	O→tcag	5→catg	M→tact			

5. Simulation

SMS simulation implementation There are several prerequisites for using the DNA cryptography Algorithm, which includes software and hardware requirements. A personal computer (PC), 8 GB RAM, a 64-bit operating system, an x64-based CPU, Android Studio software and a cellphone are required. The program prerequisites are the Windows operating system and Java/Kotlin Language.

An application was created to accomplish the suggested technique on Android using the android studio program, which is based on java/kotlin and has been upgraded to the C++ language. Java and Kotlin are the computer languages used in this study to construct encryption and decryption of communications on mobile phones using DNA (Deoxyribonucleic Acid). Kotlin combines a resource and a collection of Kotlin/Java APIs for creating mobile apps. Because it is a java/kotlin emulator, applications must have the extension Jar,.java, .kt, whereas Android phones have the extension apk and the computer must have a java platform installed: Java SE/JRE (Java Runtime Environment) and android studio.

Our Proposed solution is implemented as an app for an Android smartphone. To implement the proposed solution on Android, a chat application was created using the android studio software, which is based on the java / kotlin programming language. The system's main menu is a home screen, as seen in Figure 2. The encryption of communication might be carried out by pressing a send button, which will prompt the user to enter the message, as seen in Figure 3. we have created real world android based chat application, along with chat application in this paper we have implemented Our proposed algorithm so that the encryption and decryption process work. When we write the message and click the send button, our message goes to the receiver side in encrypted form.

When the message has been sent to the user the attacker node also receives the message as shown in Figure 4. In Figure 5 the encrypted message is sent to the receiver side. while a similar window for decryption can be shown in Figure Simulations are shown to test the successful implementation of encrypting messages with varying lengths on a sender's mobile phone. The message text was encrypted using the DNA encryption technique during testing. The encrypted communications were successfully decoded on the receiver's cell phones using DNA decryption and the original message texts could be read.

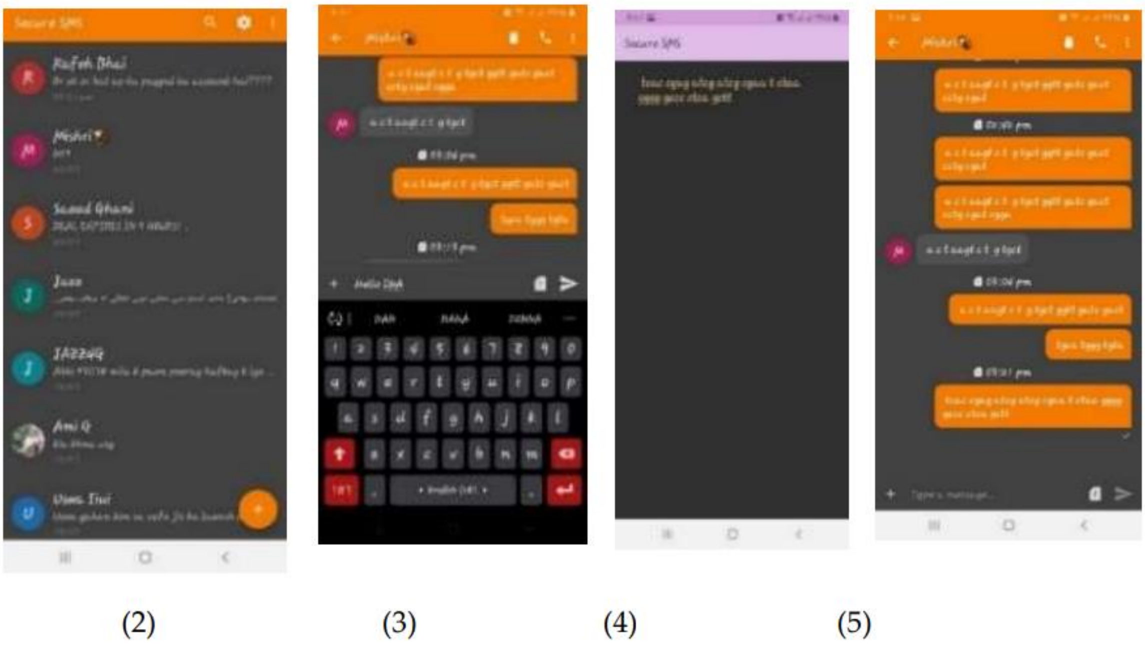

Figure 2. display the Main menu, (3) display the message typing screen, (4) Attacker node and (5) Encrypted text**.**

6. Results and Discussion

The analyses are carried out to determine whether the ECSS scheme is improved to present approaches and whether the ECSS scheme is reasonable for Mobile security or not. In the android studio environment, the proposed approach technique is examined. The SMS is encrypted using a DNA-based technique. To encrypt the text and avoid network interference, encryption stages are computed. The encryption and decryption time consumption is used to calculate the experimental outcomes in Android Studio.

6.1. Attacks

6.1.1. Plaintext Attack

The known-plaintext assault is an assault demonstrated for cryptanalysis where the aggressor has to get to both the plaintext and its scrambled form. These can be utilized to uncover and assist mystery data such as mystery keys and code books. The intruder has duplication of the ciphertext and plaintext in question and is trying to retrieve the encryption keys. Once the intruder has a portion of plaintext and ciphertext, he attempts to analyze the connection between the unencrypted and the ciphertext. This is a relatively easy cryptographic attack.

When the user provides data for encryption, the attacker wraps the plaintext. Using this known portion of data, the attacker attempts to reconstruct the encryption technique, which is subsequently employed in the decryption. The proposed solution will avoid this plaintext attack since the user does not communicate simple plaintext across the network. Plaintext always utilized a key generated followed by a process. As a result, the attacker is unable to decode the data.

$$P(E(C\ (S,R))) = A(C) + R(E) \qquad (2)$$

In equation 1, P is the plaintext that was not originally sent over the wireless channel. Using the encryption technique E, the original data is encrypted and turned into ciphertext C. Sender S sends encrypted data to receiver R. As a result, attacker A does not obtain original information from ciphertext C, which is exclusively accessible to recipient R as shown in Figure I.

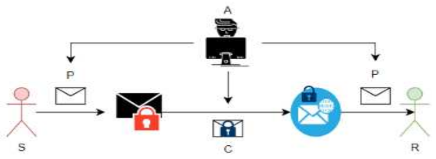

Figure I. Plaintext attack**.**

6.1.2. Dictionary Attack

A dictionary attack is a type of assault in which the encoded phrases are very similar to terms found in a dictionary, making it easy for hackers to break the plain text. Because the user does not transfer basic plaintext across the network, the proposed method avoids this Dictionary attack. Plaintext always used key generate, then process. The attacker is thus unable to decrypt the data shown in Figure II.

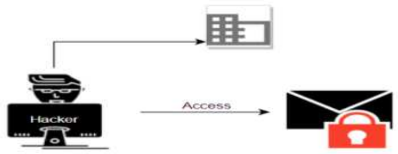

Figure II. Dictionary Attack.

6.1.3. Man in the Middle Attack

This attack occurs when the hacker transmits both sides' data from its stream and, if necessary, modifies the content. At the start of the session, the attacker locates the users and opens two separate sessions with the users, assuming the users are directly linked. The data of the users are routed through the attacker's channel and supplied after alteration. The proposed method triumphs over the onslaught.

6.2. Computational Time

The computational time is the amount of time it takes an algorithm to compute its results. The computational time is an important output metric for an encryption strategy that shows how many times a specific operation is executed.

6.3. The Computational Time of Encryption

The DNA-based algorithm's computing time is determined. It employs an effective approach for encrypting messages to prevent network interference difficulties. The encryption time taken by the suggested technique to compute efficient results is shown in Figure 6. When the number of input information bits increases, so does the time required by the method; nevertheless, in the given scenario, the time complexity for SMS encryption/decryption is small Figure 7.

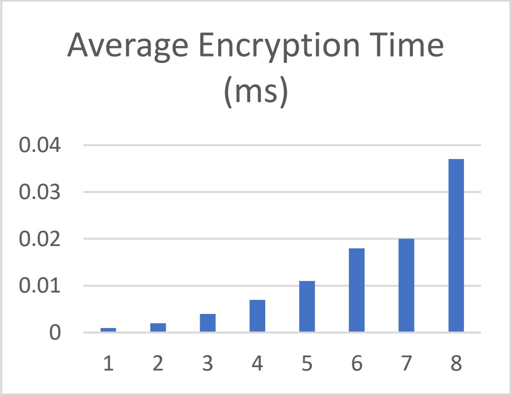

Figure 6. Average Encryption Time.

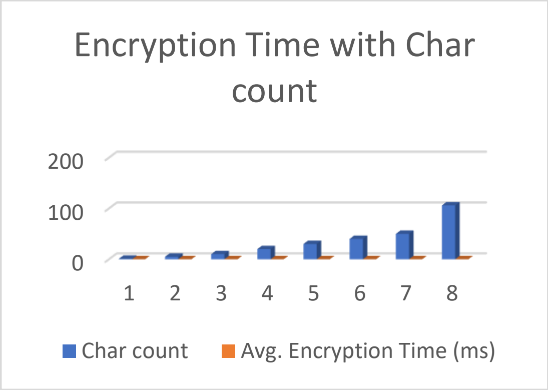

Figure 7. Char count with Encryption Time.

6.4. The Computational Time of Decryption

On the receiver side, the computational time required by the proposed DNA-based technique to decrypt the ciphertext is determined. To decode ciphertext into original data, it employs the ECSS method. Figure 8 shows how raising the output data bytes increases the decoding time of ciphertext in milliseconds Figure 9. The overall time required is quite low and the suggested approach is quick.

doi:10.20944/preprints202401.0495.v1

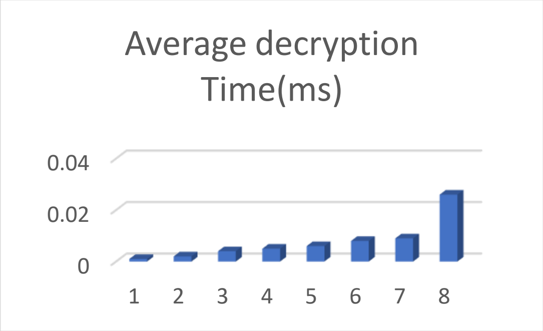

Figure 8. Average Decryption Time.

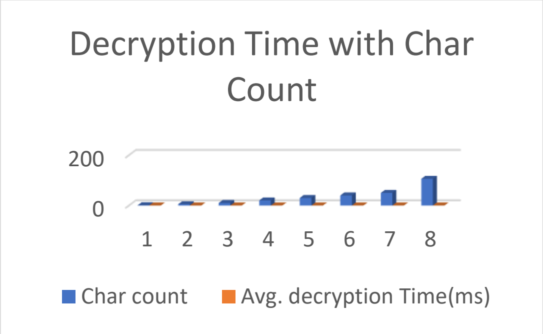

Figure 9. Char count with Decryption Time.

In **Figure** 10 different cipher size are used for encryption. In **Figure** 11 The test results suggest that all communications are delivered between sender and receiver, are 100% innovative and better.

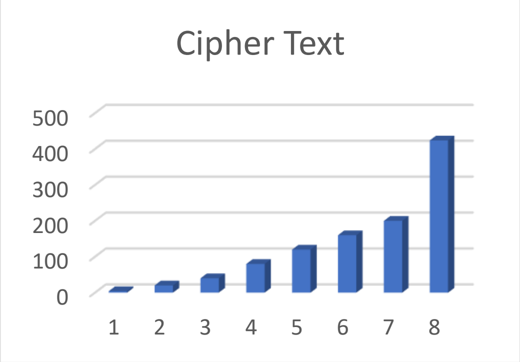

Figure 10. Cipher text

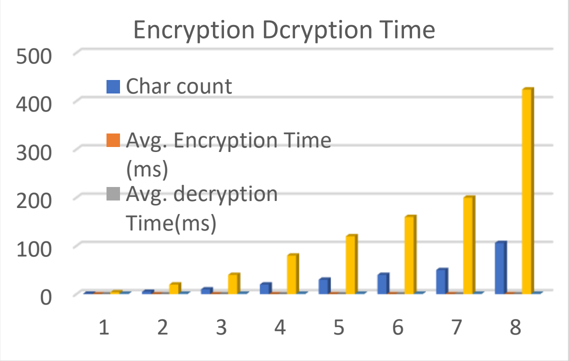

Figure 11. Encryption/Decryption Time with Cipher text and Accuracy.

Cryptosystems take an average of 0.037 and 0.026 sec, correspondingly to Table 3.

Table 3. Encryption/Decryption Time.

Char count	Avg. Encryption Time	Avg. decryption Time	Cipher Text	Accuracy
1	0.001	0.001	4	100%
5	0.002	0.002	20	100%
10	0.004	0.004	40	100%
20	0.007	0.0044	80	100%
30	0.011	0.006	120	100%
40	0.018	0.008	160	100%
50	0.02	0.009	200	100%
106	0.037	0.026	424	100%

6.5. Comparison Analysis

To measure the algorithm's efficiency, a comparison analysis of numerous methodologies with the ECSS scheme is undertaken. In **Figure** 12 and **Figure** 13, The computational time complexity of encryption and decryption is smaller than that of the comparable techniques AES[9] and ECC [12].In **Figure** 14 the ECSS scheme is compared with the other three techniques with 160 char count. In **Figure** 15 the ECSS-based approach required less computing time than existing strategies.

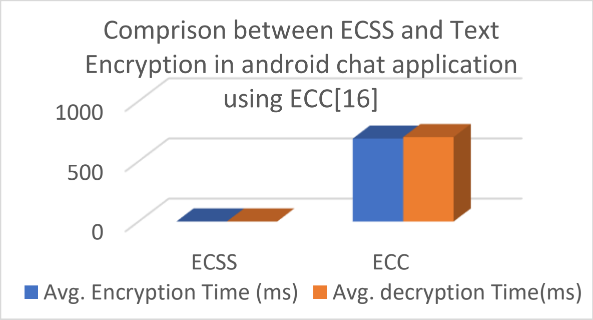

Figure 12. Comparison between ECSS and [16] with 25 char.

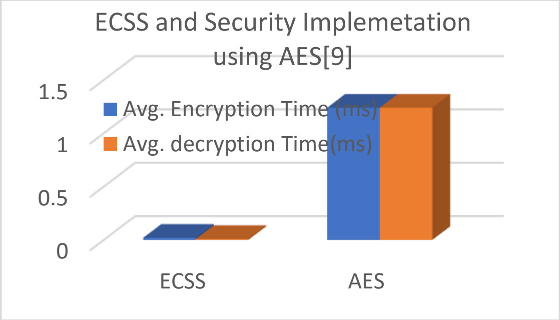

Figure 13. Comparison between ECSS and [9] with 50 char.

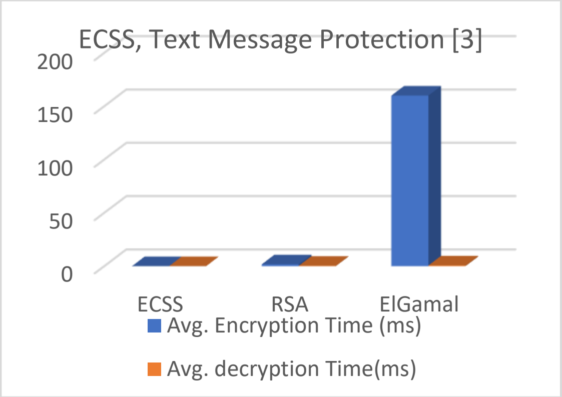

Figure 14. ECSS,[3].

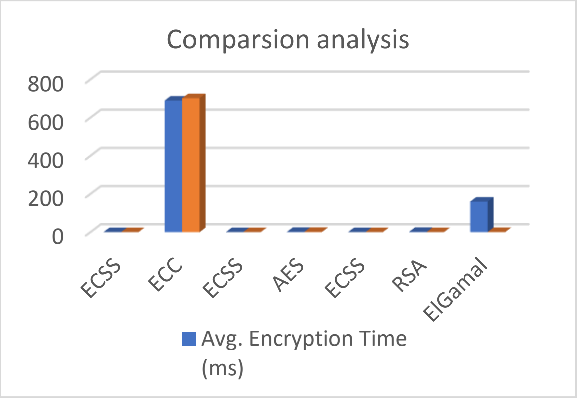

Figure 15. Comparison analysis of algorithms.

Table 4 indicates that the encoding is affected by the length of the key; as the length increases, so does the time required for encryption. In this paper, this study compares the time of encryption and decryption we other techniques that were used previously for the creation of secure encryption and decryption processes of a chat application. So that the average encryption and decryption speed of the message is 0.007 for 25 char and 0.011 for 160 char count as shown in Table 4.

Table 4. Performance Data**.**

	ECSS	ECC	ECSS	AES	ECSS	RSA	ElGamal
Char count	25	25	50	50	160	160	160
Avg. Encryption Time	0.007	689.2	0.02	1.24	0.011	1.675	160
Avg. decryption Time	0.0048	701.8	0.009	1.24	0.038	0.38	0.68

7. Conclusions

This research shows our proposed system for developing an Android-based chat application with end-to-end encryption. A chat program that uses the ECSS algorithm for text encryption and decryption. This study demonstrates that the ECSS algorithm can be implemented and has competitive performance in terms of both time and accuracy. Since this paper implemented the ECSS algorithm, our security has greatly improved. Because this paper employed a random key generation mechanism, it is immune to dictionary and plain text assaults. Because this research is providing DNA values for encryption and decryption, it is difficult for adversaries to crack. The experiment results reveal that the suggested encryption technique has excellent avalanche resistance.

References

1. DataReportal, "Digital 2020 Pakistan (January 2020) v01," 04:10:26 UTC. Accessed: Jul. 19, 2022. [Online]. Available: https://www.slideshare.net/DataReportal/digital-2020-pakistan-january-2020-v01
2. "Mobile Operating System Market Share Pakistan," StatCounter Global Stats. https://gs.statcounter.com/os-market-share/mobile/pakistan (accessed Jul. 21, 2022).
3. A. Starikovskiy, A. Zhgilev and N. Shevchenko, "Text Messages Protection System," 8th Annual International Conference on Biologically Inspired Cognitive Architectures, BICA, Moscow, Russia 2017, vol. 123, pp. 457–466, 2018.
4. A. F. Ramdhansya, E. Ariyanto and H. H. Nuha, "Implementasi Advanced Encryption Standard (AES) Pada Sistem Kunci Elektronik Kendaraan Berbasis Sistem Operasi Android Dan Mikrokontroler Arduino," Seminar Nasional Informatika 2014 Neliti.com. [Online]. Available: https://media.neliti.com/media/publications/175408-ID-implementasi-advanced-encryption-standar.pdf, 2019.
5. M. W. Khan, "SMS Security in Mobile devices: A Survey," International Journal Advanced Networking and Applications vol. 05, no. 02, pp. 1873-1882, 2013.
6. O. S. Sitompul, N. H. Pasaribu and E. B. Nababan, "Hybrid RC4 and Affine Ciphers to Secure Short Message Service on Android," in 2018 Third International Conference on Informatics and Computing (ICIC), IEEE, Palembang, Indonesia, pp. 1–6, 2018.
7. S. Tahira, A. Ullah, H. Ashraf and M. Sher, "Efficient security associations establishment using IPSec in IMS after handover in NGMN." Journal of Internet Technology vol.20, no.2, pp.359-367,2019.
8. H. Ashraf, A. Ullah, S. Tahira and M. Sher, "Efficient certificate based One-pass Authentication Protocol for IMS." 1134 Journal of Internet Technology vol.20, no. 4, pp.1133-1143,2019.
9. D. Nurmalasari, E. Mulyana and M. Irfan, "Security implementation of the Internet of Things using the Advanced Encryption Standard (AES) Algorithm," in 2019 IEEE 5th International Conference on Wireless and Telematics (ICWT), IEEE, Yogyakarta, pp. 1–4, 2019.
10. B. B. Ahamed and M. Krishnamoorthy, "SMS Encryption and Decryption using Modified Vigenere Cipher Algorithm," Journal of the Operations Research Society of China, vol. 10, pp.1-14, 2020.
11. D. Natanael and D. Suryani, "Text Encryption in Android Chat Applications using Elliptical Curve Cryptography (ECC)," The 3rd International Conference on Computer Science and Computational Intelligence (ICCSCI 2018), vol. 135, Indonesia, pp. 283–291, 2018.
12. A. P. U. Siahaan, "An overview of the RC4 Algorithm," IOSR Journal of Computer Engineering (IOSR-JCE) vol. 18, pp 67-73,2016.
13. K. Rathi, U. Karabiyik, T. Aderibigbe and H. Chi, "Forensic analysis of encrypted instant messaging applications on Android," in 6th International Symposium on Digital Forensic and Security (ISDFS), IEEE, Turkey, pp. 1–6,2018.
14. O. A. Dawood and O. I. Hammadi, "An analytical study for some drawbacks and weakness points of the AES Cipher (Rijndael Algorithm)," A Scientific Quarterly Refereed Journal, vol. 2, no. 2, pp.111-118. 2017.
15. "Snapshot." Accessed: Jul. 19, 2022. [Online]. Available: https://en.wikipedia.org/wiki/Advanced_Encryption_Standard
16. N. H. UbaidurRahman, C. Balamurugan and R. Mariappan, "A novel DNA computing based encryption and decryption algorithm," International Conference on Information and Communication Technologies (ICICT 2014), vol. 46, India, pp. 463–475, 2015
17. H. Al-Mahdi, M. Alruily, O. R.Shahin and K. Alkhaldi, "Design and analysis of DNA encryption and decryption technique based on asymmetric cryptography system," (IJACSA) International Journal of Advanced Computer Science and Applications, vol. 10, no. 2, 2019.
18. Z. K. Abdalrdha, F. N. Abbas and I. H. AL-Qinani, "Subject review: SMS encryption for android mobile using the encryption algorithm," International Journal of Engineering Research and Advanced Technology (IJERAT), vol. 05, no. 10, pp. 01–08, 2019.
19. E. Vidhya, S. Sivabalan and R. Rathipriya, "Hybrid key generation for RSA and ECC," Fourth International Conference on Communication and Electronics Systems (ICCES), IEEE, India, pp. 35-40 2019.
20. A. H. Ali and A. M. Sagheer, "Design of an android application for secure chatting," International Journal of Computer Network and Information Security, vol. 9, no. 2, pp. 29–35, 2017.
21. T. Mandge and V. Choudhary, "A DNA encryption technique based on matrix manipulation and secure key generation scheme," 2013 International Conference on Information Communication and Embedded Systems (ICICES), Chennai, India, pp. 47-52, 2013.

22. E. M. S. Hossain, K. M. R. Alam, M. R. Biswas and Y. Morimoto, "A DNA cryptographic technique based on dynamic DNA sequence table," 19th International Conference on Computer and Information Technology (ICCIT), Dhaka, Bangladesh, pp. 270-275, 2016.
23. F. E.Ibrahim, M. I. Moussa and H. M. Abdalkader, "A Symmetric Encryption Algorithm based on DNA Computing," International Journal of Computer Applications, vol. 97, no. 16, pp. 41–45, 2014.
24. M. Imran, M. Rashid, A. Raza Jafri and M. Najam-ul-Islam, "ACryp-Proc: Flexible Asymmetric Crypto Processor for Point Multiplication," IEEE Access, vol. 6, pp. 22778–22793, 2018.
25. A. N. Borodzhieva and P. K. Manoilov, "Training module with graphical user interface for encryption and decryption using affine ciphers applied in cryptosystems," 2014 IEEE 20th International Symposium for Design and Technology in Electronic Packaging (SIITME), Bucharest, Romania, pp. 281-286, 2014.
26. S. Shukla and P. Verma, "Implementation of Affine Substitution Cipher with Keyed Transposition Cipher for Enhancing Data Security," International Journal of Advanced Research in Computer Science and Software Engineering 4, no.1, pp. 236-241, 2014.
27. D. Luciano and G. Prichett, "Cryptology: from Caesar Ciphers to public-key Cryptosystems," Mathematical Association of America, vol. 18, no. 1, pp. 2–17, Jan. 1987.
28. M. Biswas, and N. Sakeef and H. Siddique, "Exploring network security using Vigenere Multiplicative cipher encryption and implementation," Research Square, 2020, doi:10.21203/rs.3.rs-58356/v1.
29. R. Rifki, A. Septiarini and H. Rahmania, "Cryptography using random RC4 stream cipher on SMS for Android-based smartphones," International Journal of Advanced Computer Science and Applications(IJACSA), vol. 9, no. 12, 2018, doi: 10.14569/IJACSA.2018.091214.
30. P. Jindal and B. Singh, "RC4 encryption-A literature survey," International Conference on Information and Communication Technologies (ICICT 2014), vol. 46, Kochi, pp. 697–705, 2015.
31. N. Kumar and P. Chaudhary, "Performance Evaluation of Encryption/Decryption Mechanisms to Enhance Data Security," Indian Journals of Science and Technology, vol. 9, no. 20, pp.1-10, 2016.
32. H. Al-Mahdi, O. R.Shahin, Y. Fouad and K. Alkhaldi, "Design and analysis of DNA Binary Cryptography Algorithm for Plaintext," International Journal of Engineering and Technology (IJET), vol. 10, no. 3 , pp. 699–706, Jun. 2018.
33. R. Rayarikar, S. Upadhyay and P. Pimpale, "SMS Encryption using AES Algorithm on Android," International Journal of Computer Applications, vol. 50, no. 19, pp. 12–17, 2012.
34. A. Almusaylim, Z., Jhanjhi, N. Z., & Alhumam, A. (2020). Detection and mitigation of RPL rank and version number attacks in the internet of things: SRPL-RP. Sensors, 20(21), 5997.
35. Humayun, M., Jhanjhi, N. Z., Talib, M. N., Shah, M. H., & Suseendran, G. (2021). Cybersecurity for Data Science: Issues, Opportunities, and Challenges. Intelligent Computing and Innovation on Data Science: Proceedings of ICTIDS 2021, 435-444.
36. Kok, S. H., Azween, A., & Jhanjhi, N. Z. (2020). Evaluation metric for crypto-ransomware detection using machine learning. Journal of Information Security and Applications, 55, 102646.
37. Shafiq, M., Ashraf, H., Ullah, A., Masud, M., Azeem, M., Jhanjhi, N. Z., & Humayun, M. (2021). Robust Cluster-Based Routing Protocol for IoT-Assisted Smart Devices in WSN. Computers, Materials & Continua, 67(3).
38. Hussain, K., Hussain, S. J., Jhanjhi, N. Z., & Humayun, M. (2019, April). SYN flood attack detection based on bayes estimator (SFADBE) for MANET. In 2019 International Conference on Computer and Information Sciences (ICCIS) (pp. 1-4). IEEE.
39. Lim, M., Abdullah, A., Jhanjhi, N. Z., & Supramaniam, M. (2019). Hidden link prediction in criminal networks using the deep reinforcement learning technique. Computers, 8(1), 8.
40. Kumar, T., Pandey, B., Mussavi, S. H. A., & Zaman, N. (2015). CTHS based energy efficient thermal aware image ALU design on FPGA. Wireless Personal Communications, 85, 671-696.
41. Verma, S., Kaur, S., Rawat, D. B., Xi, C., Alex, L. T., & Jhanjhi, N. Z. (2021). Intelligent framework using IoT-based WSNs for wildfire detection. IEEE Access, 9, 48185-48196.
42. Khalil, M. I., Jhanjhi, N. Z., Humayun, M., Sivanesan, S., Masud, M., & Hossain, M. S. (2021). Hybrid smart grid with sustainable energy efficient resources for smart cities. sustainable energy technologies and assessments, 46, 101211.
43. Diwaker, C., Tomar, P., Solanki, A., Nayyar, A., Jhanjhi, N. Z., Abdullah, A., & Supramaniam, M. (2019). A new model for predicting component-based software reliability using soft computing. IEEE Access, 7, 147191-147203.
44. Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023, April). Yolo-based deep learning model for pressure ulcer detection and classification. In Healthcare (Vol. 11, No. 9, p. 1222). MDPI.
45. Aldughayfiq, B., Ashfaq, F., Jhanjhi, N. Z., & Humayun, M. (2023). A Deep Learning Approach for Atrial Fibrillation Classification Using Multi-Feature Time Series Data from ECG and PPG. Diagnostics, 13(14), 2442.

46. Hussain, S. J., Ahmed, U., Liaquat, H., Mir, S., Jhanjhi, N. Z., & Humayun, M. (2019, April). IMIAD: intelligent malware identification for android platform. In 2019 International Conference on Computer and Information Sciences (ICCIS) (pp. 1-6). IEEE.
47. Sennan, S., Somula, R., Luhach, A. K., Deverajan, G. G., Alnumay, W., Jhanjhi, N. Z., ... & Sharma, P. (2021). Energy efficient optimal parent selection based routing protocol for Internet of Things using firefly optimization algorithm. Transactions on Emerging Telecommunications Technologies, 32(8), e4171.
48. Wassan, S., Chen, X., Shen, T., Waqar, M., & Jhanjhi, N. Z. (2021). Amazon product sentiment analysis using machine learning techniques. Revista Argentina de Clínica Psicológica, 30(1), 695.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.