Recurrence relation and DNA sequence: A state-of-art technique for secret sharing

Anirban Bhowmik[1], Sunil Karforma[2], Joydeep Dey[3]
[1,3]Department of Computer Science, MUC Women's College, Purba Barddhaman, WB, India
[2]Department of Computer Science, The University of Burdwan, Purba Burdwan, WB, India

Article Info

Article history:

Received Sep 8, 2020
Revised Jan 6, 2021
Accepted Feb 24, 2021

Keywords:

DNA sequence
Message digest
Recurrence relation
RSA
Secret sharing
Session key
Symmetric key

ABSTRACT

During the transmission over the Internet, protection of data and information is an important issue. Efficient cryptographic techniques are used for protection but everything depends on the encryption key and robustness of encryption algorithm. Threshold cryptography provides the development of reliable and strong encryption and key management machine which can reconstruct the message even in the case of destruction of some particular numbers of shares and at the opposite the data cannot be reconstructed unless an allowable set of shares are been gathered. The earlier techniques available in literature result in high computational complexity in the course of both sharing and reconstructing of message. Our method employs a brand new easy protecting technique based totally on unit matrix. The simple AND operation is used for percentage generation and reconstruction can be finished by way of easy ORing the stocks with threshold cost. We are proposing a sharing approach in conjunction with conventional cryptography technique for key control to make the key greater sturdy and for encryption we have used a session key the use of the idea of recurrence relation and DNA series Different types of experimental results confirm authenticity, confidentiality, integrity and acceptance of our technique.

Corresponding Author:

Anirban Bhowmik
Department of Computer Science
MUC Women's College
B.C. Road, Purba Bardhaman, WB, India
Email: animca2008@gmail.com

1. INTRODUCTION

The The effective and secure protection of the private keys in cryptography is a significant issue in modern era. There are many cryptography techniques [1], but there is some weakness to these techniques. The private keys should not provide to an individual because single point failure may occur. A powerful way to communicate the important thing through wireless channel securely is to use of the secret key on specific situation. A (k,n) threshold based secret sharing scheme [2, 3] can be used where the name of the game records is shared among n numbers of members such that a group of ok or greater individuals reconstruct the name of the game but no longer for less than k.

A function sharing hassle is one of the shortcomings of ideal secret sharing [4, 5] scheme. Where feature computation [6] is distributed according to mystery sharing scheme such that the character user computes the shared components and then the partial end result can be mixed to yield the final result with out disclosing the character secrets. Various feature sharing protocols are there Shamir secret sharing primarily based on polynomial interpolation, Blakley's secret sharing primarily based on hyper plane geometry and

Asmuth-Bloom based on Chinese the rest theorem [7]. Recurrence relation: Linear recurrence [8] is defined as each term of a sequence is a linear function of earlier terms. Recurrence relation is of two types: 1) linear recurrence relation, 2) linear non homogeneous recurrence relation.

Linear recurrence relation: A linear homogenous recurrence relation of degree k with constant coefficients is a recurrence relation of the form $a_n = c_1 a_{n-1} + c_2 a_2 + \dots + c_k a_{n-k}$, where c_1,c_2,…,c_k are real numbers, and $c_k \neq 0$. a_n is expressed in terms of the previous k terms of the sequence. Proposition 1: Let $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k}$ be a linear homogeneous recurrence. i) Assume the sequence a_n satisfies the recurrence. ii) Assume the sequence a'_nalso satisfies the recurrence. So, $b_n = a_n + a'_n$ and $d_n = \alpha\, a_n$ are also sequences that satisfy the recurrence. ($\propto$ is any constant).

Linear non-homogeneous recurrence: A linear non-homogenous recurrence relation [8] with constant coefficients is a recurrence relation of the form $a_n = c_1 a_{n-1} + c_2 a_{n-2} + \dots + c_k a_{n-k} + f(n)$, where c_1, c_2…,c_k are real numbers, and $f(n)$ is a function depending only on n. The recurrence relation $a_n = c_1\, a_1 + c_2\, a_2 + \dots + c_{n-k} a_{n-k}$, is called the associated homogeneous recurrence relation.

Here the concept of recurrence relation is used for random number generation which is used for key generation. In descrete mathematics the use of recurrence relation in random number generation [9-11] is a new concept. The details algorithm is given in next section. DNA sequence: DNA is Deoxy Neuclic Acid [12] that is the start line of all life. DNA molecules contain strands of nucleotides that are: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T). The DNA molecules comprise double helix structure combinining the complementary strands A to T and G to C. The mixture of bits represents these 4 bases as shown in Table 1. Incorporating of the features of DNA in cryptography is a singular concept as for instance the DNA encryption technique. In many methods we will use the DNA traits. The plain textual content can be converted into ASCII codes after which into binary code, that can then be replaced through DNA bases A, T, G and C [13, 14].

Table 1. DNA encoding

Bits	Base
00	A
01	T
10	G
11	C

2. RELATED BACKGROUND WORKS

Shamir's secret sharing scheme: Shamir's secret sharing scheme is based on (k, n) threshold primarily based secret sharing technique [15]. In this scheme a $(k-1)$ degree polynomial is important. The polynomial feature of order $(k-1)$ is built as follows - $f(x) = (p_0 + p_1 x_1 + p_2 x_2 + p_3 x_3 + \dots + p_{k-1} x_{k-1}) mod\ m$. Where $p_0$the secret and m is is a prime number and all other coefficients are selected randomly from secret. Each of the n shares is a pair (x_i, z_i) of numbers satisfying $f(x_i) = z_i$ and $x_i > 0, 1 \le i \le n$ and $0 < x_1 < x_2 < x_3 < \ldots < x_k \le m - 1$. Given any k shares, the polynomials are uniquely determined and hence the secret p_0 can be computed via Lagrange's interpolation.

Blakey's secret sharing scheme: Blakey's secret sharing scheme used geometry to remedy secret sharing trouble [3]. The secret message is a factor in a k- dimensional space and n stocks are affine hyper planes that intersect on this point. The set answer $y = (y_1, y_2, y_3, \dots, y_k)$ to an equation $p_1 y_1 + p_2 y_2 + p_3 y_3 + \dots + p_k y_k = b$ forms an affine hyper plane. The intersection point is obtained by finding the intersection of any k of these planes.

Asmuth-Bloom's secret sharing scheme: Asmuth-bloom's secret sharing scheme [2] shares a secret many of the individual events using modular arithmetic and reconstruction it with the aid of Chinese remainder theorem (CRT). Above all of the secret sharing schemes are appeared as a great secret sharing scheme because amalgamation of (k-1) shares doesn't reveal any facts approximately the secret. In key based secure threshold cryptography, initially a 16-byte digest string is generated from given variable length key using MD5 and this key is shared.

3. OUR CONTRIBUTION

In this article the three novel aspects of our work are – (i) the use of recurrence relation in random number generation (ii) the use of DNA sequence for non linearity (iii) the use of unit matrix in mask generation algorithm. This mask generation process is a new approach in cryptography and it is very easy than Shamir's secret sharing scheme or Blakey's secret sharing scheme with respect to complexity. Besides we have included symmetric key and hash function based message authentication and RSA based user authentication. For session key generation we have used the concept of recurrence relation and DNA sequence. The share generation is done on generated mask. All the steps are described below by a proper algorithm. A case study is also given for clear understanding of proposed scheme.

4. OUR PROPOSED TECHNIQUE

The summary of our proposed technique is given through an algorithm which contains five modules. Each module is described via an algorithm.

```
Algorithm:
Input: seed values, plain text, symmetric key.
Step1: call mask_Generation ( ).    // Mask generation algorithm.
Step2: call SKG ( ).                // Session key generation.
Step3: call enc_msg ( ).            // plain text encryption.
Step4: call share_Generation ( )    // share generation & transmission file creation
Step5: call reconst_msg ( ).        // decryption and reconstruction of message.
        End
```

4.1. Mask generation module

Our proposed work is largely a depending on protecting the pre-defined knumber of shares on the message or secret data and then to carry out OR operation on the pre-defined k variety of shares to regenerate the unique transmitted message or secret information. The secret data may be considered as an image, audio, video or any text file. Our initial task is to decompose the file of any size into k number of shares. On the receiver end, if we perform bitwise OR operation upon k number of shares then only the original data will be reconstructed, not even upon $(k-1)$ shares. Every share must have some missing bits and hence those missing bits can be replaced by k shares exactly.

Step1: From n number of total recepients we can choose any k number of recipients to send message $(m < k)$.

Step2: A unit matrix of order kxk is taken.

Step3: The mask matrix is generated by shuffling all the rows of unit matrix.

Step4: The mask matrix is used for share generation.

4.2. Session key generation module

In this module we have generated session key for data encryption. Here Non-Homogeneous Recurrence Relation formula is used for the session key generation. At firat session key is XORed with symmetric key and then divide into k number of shares using mask matrix and then each share of session key is transmitted to receiver end as an attachment of each share of message. The size of session key is same as the size of symmetric key.

Algorithm:

Inputs: - seed values, coefficient value and non homogeneous recurrence equation.

Outputs: - session key

Method:

```
1. Set i, j, m, n, f, lr as integer and $a[m], r[n], c[m]$as integer array.
2. $n <-$ total random number.
3. $m <-$ total no.of coefficient in non-homogeneous recurrence relation.
4. For $i = 0 \ to \ m$
5. $c[i] <-$ get_coeff () and $a[i] <-$get_seedVal ()
           end for
6. $lr <-$ get_largestPrimeFact ($a[2] \ xor \ c[3]$). {/* a[2] and c [3] are chosenby the user.*/}
7. $a[0] <- a[0] \ xor \ lr$.
8.  for $i = 1 \ to \ m$
9. $a[i] = (a[i] xor \ a[i-1]) \ xor \ lr$
   end for
10. for $i = 3 \ to \ n$
11. $a[i] <- get\_val \ (rec\_Funct \ (i))$
12. if ($a[i] < 0$)
13. $a[i] <- a[i]$
     end if
14. $r[i] <- a[i]$
15.  f<-(((($a[i] xor \ c[1]) xor \ c[3]) xor \ c[5]) xor \ldots c[m]$)
16. $a[i] <- f$
```

```
    end for
if ($n >= 3$)
 $r[n] < -$get_shuffle($r[n]$) // $r[n]$ represent the session key.
$sesnk[]$← get_DNAsequen ($r[n] XOR\ symmertic\ key$)
    end if
End
```

DNA Sequence Module (get_DNAsequence ()): It takes random sequence generated from RC as parameter vxxalues.

1. Each integer is divided into four bits.
2. DNA sequence is generated by taking two bits together and values of Table 1.
3. Thus, we get a sequence of A, T, C, G in any order which is treated as Session key.

4.3. Encryption and share generation module

Algorithm:
Inputs: - session key, symmetric key, plain text and total number of recipient.
Outputs: - share generation and transmission file.
Method:

1. set plaintext[], ciphertxt[] , share_msg[][], share_sessionkey[][], and share_hash[][]as Character array and k as integer.
2. ciphertxt [k] ← call XoR_OP (plaintext[k], session key) and k← no.of share.
3. share_sessionkey [k] [k] ←call share_Generation (mask matrix[k] [k], session key).
4. share_msg [k] [k] ←call share_Generation (mask matrix[k] [k], ciphertxt [k] [k]).
5. share_hash [k] [k] ←call share_Generation (maskmatrix[k][k],hash(symmetric key)).
6. Create transmission file for each recipient.
7. The transmission file is sent to the recipient end using public key of each recipient.
8. End.

The transmission file [16] is created using message share, session key share and hash value share. The Structure is given below. This transmission file is termed as message.

SHARE OF CIPHER TEXT	SHARE OF SESSION KEY	HASH VALUE SHARE(PADDING)

This transmission file is generated for each share. For hash value generation, MD5 hash function on symmetric key is used as hash value. Each share of message is generated by using AND operation between each row of mask matrix and message. Each share is transmitted to recipient end through RSA.

4.4. Reconstruction of encrypted message (decryption module)

Algorithm:
Input: - public keys of n number of recipients and their shares.
Output: - Original secret message
Method: -

1. k number of recipients should join in key reconstruction using their individual shares.
2. The recipients will decrypt their individual share by their private keys. This provides the user Authentication.
3. At first hash vlaue of symmetric key is generated for message authentication and to get the hash value back, it is necessary to bit by bit ORed of each k number of shares of recipients.
4. After checking the authentication both user and message, the session key is generated by using OR operation on bit by bit of each k number of shares.
5. k Numbers of recipient generate message or data by using bit by bit OR operation on the shares.
6. The decryption is done on cipher text to get the plain text by using session key.

5. A CASE STUDY

A snapshot which contains a set of seven recepients and out of seven five recipients are chosen to send message. Thus, the threshold value is five and it is given in the following tabular format. Now according to mask generation algorithm first a unit matrix of order 5x5 is chosen and then shuffle operation is done 5 times.

1	0	0	0	0
0	1	0	0	0
0	0	1	0	0
0	0	0	1	0
0	0	0	0	1

Unit matrix of 5x5

After shuffle operation we get the following matrix. Shuffle operation may be *n* times, where *n* less than number of rows of matrix.

0	1	0	0	0
1	0	0	0	0
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0

Now the shares are shown in Table 2.

Table 2. Snapshots of shares

0	1	0	0	0	Share No. 1
1	0	0	0	0	Share No. 2
0	0	0	0	1	Share No. 3
0	0	0	1	0	Share No. 4
0	0	1	0	0	Share No. 5

Let the message or plain text be *A7BC1DE3FH2G*, the session keyAC3GE8RD is and the symmetric key be *BG65$gAdS.*

Let Sender's key pair	public (137, 83)		
Recipient- 1 key pair	public (97, 73)	Recipient -2 key pair	public (197, 41)
Recipient -3 key pair	public (103, 173)	Recipient -4 key pair	public (173, 41)
Recipient -5 key pair	public (97, 23)		

Now after XOR operation between plain text and session key we get the cipher text as DF2h9RaC75Tb.
Now using a hash algorithm (MD5) the digest of the symmetric key is c7c52f2bbab358795947dfbd27e5d63b**.**
Let MSG be the message format which is to be transmitted to the recipient end via wireless channel.

	Encrypted data	session key	Padding
The MSG:	DF2h9RaC75Tb	*ATGCTTACG*	c7c52f2bbab358795947dfbd27e5d63b

Now using the AND operation between the values of each row of Table 2 and MSG and as a result we get following shares.

DF2h9 RaC7 5Tb *ATGCTTACG* c7c52f2bbab358795947dfbd27e5d63b
01000 01000 01000 01000 01000 01000 01000 01000 0100001000010

and we get the share1 as follows.

1st share: 0F0000R0000T000G0000C000050000b0000700007000030
Similarly, we get the following shares.
2nd share: 00f00600000f0TGA0000b0009f000d0000090000a00000d00
3rd share: 0D20200d8f0TA0A0052f0b00b008090e0000500008000070
4th share: 0000000d80T0A0G00C2f00b0b00809060000f000050000d0
5th share: 0020260080A0G0T00G2020b0b0507950000c000040000d0

Next the above shares are encrypted by the corresponding individual public key of the each recipient and send them. Now in the recipient end, hash value of symmetric key is generated by using the OR operation among five shares. This hash value i.e., padding field is used for message authentication purpose in secret sharing. Next, session key is generated using the OR operation among five shares. This session key is used for decryption purpose. At last, the cipher text is reconstructed by OR operation and then the palin text are generated by using XOR operation between session key and cipher text and their private keys.

	CipherTxt	session key	Padding
1st share:	0D00060d0 f0b	0T0000A00	a0300909070f0d070000a0000f0d000c
2nd share:	00f00600c000	A0000T000	0000b00b3500009000000f0000f0d000
3rd share:	0D20200d00a0	0000T0000	052f0b00b00809000000c0a00d0d000c
4th share:	g00e000d0a00	000C0000G	052f00b0b00809090000a0000f0d000c
5th share:	00202600b00d	00G0000C0	c02020b0b05079500000a0000f0d000c

The plain text: A7BC1DE3FH@G.
The above case study proves the novelty of our proposed scheme.

6. RESULTS AND ANALYSIS OF OUR PROTOCOL

In this section, simulation results of the proposed technique are presented. All the programs and calculations are done in a machine with following configurations. Table 3 shows the configuration details. In our experiments, different types of files are used as plain text. Our result section is divided into four parts: (i) Analysis of mask generation (ii) Analysis of key strength (iii) Analysis of encryption technique [17, 18].

Table 3. Machine description

Computer	Lenovo G80.
Processor	Intel® Pentium® CPU B950@210GHz
RAM	2GB
Compiler	Turbo C
Disc Drive	SA 9500325AS ATA
Operating System	Windows 7 Ultimate (32 Bits)

6.1. Comparative analysis of the mask generation algorithm

Here we've got used a text message as secret. But our proposed approach is likewise similarly applicable for any binary record consisting of Image (.Bmp), audio and many others. In secret sharing [4] for n stocks with threshold value k duration of each mask is ${}_{k-1}^{\;n}C$ where we have ${}_{k-2}^{n-1}C$ zeros and ${}_{k-n}^{n-1}C$ ones. Then each share contains ${}_{n-k}^{n-1}C$ number of bytes for ${}_{k-1}^{\;n}C$ number of bytes of secret message. The original secret message can be reconstructed by only k number of collating shares: otherwise message cannot be reconstructed. Because fewer shares cannot reconstruct the original header, thus we cannot have either right key (K) or the information to construct the correct masking pattern.

Our proposed approach has used best unit matrix of order k in area of permutation and aggregate of preceding approach. It can declare to be a Perfect Secret Sharing (PSS) method [19] as well as clean with recognize to other. Here all generated shares are compressed and comprise partial secret information in encrypted shape that provides extra safety to the name of the secret message. Only when allowable set of shares comes collectively, then simplest the authentic secret message is reconstructed. Now in case of big values of n and k, previous secret sharing method suggests greater complexity than our technique. Thus with recognize to time complexity and computational complexity our mask generation method is higher than previous techniques. Here two algorithms are used one for unit matrix era and different for suffllling the rows of unit matrix.

6.2. Analysis of key strength

In our protocol shared data is very sensitive with respect to the secret key value. The session key is used as encryption key and it is generated using recurrence formula. Different types of experimental results prove the strength of our session key.

Randomness test & entropy test on session key: - Here the session secret is generated from recurrence relation and DNA collection. Now to check the randomness of session key we use some stylish techniques which include frequency test [10], entropy [20, 21].

Frequency test: -The frequency check is the maximum simple check for randomness checking. The purpose of this scheme is to determine whether or not or not the range of 1's and 0's in a series is approximately similar to might be expected for a genuinely random sequence. Mathematical Structure of the Test:
Frequency (n), where n is the length of bit string.

$\mathcal{E}$: the sequence of bits which are generated by RNG or PRNG.
S_{obs}: the absolute value of the sum of the X_i (where $X_i = 2\mathcal{E} - 1$) is the sequence divided by the square root of the length of the sequence.

1) Conversion to ± 1: The zeros and ones of the input sequences ($\mathcal{E}$) are converted to values of -1 and +1 and are added together to produce$S_n = X_1 + X_2 + \cdots + X_n$, where X_i= $2\mathcal{E}_i - 1$.
2) Compute the test static $S_{obs} = ABS\ (S_n)/\sqrt{n}$.
3) Compute $P - value = erfc\,(S_{obs}/\sqrt{2})$.
4) If $P - value \geq 0.01$ then the conclusion is that the sequence is random and if $P - value < 0.01$ then the sequence is not random.

The following Table 4 and Figure 1 show the details of frequency test result.

Table 4. Frequency test result

Session key size (bits)	Frequency test result of our technique.	Frequency test result of PRNG()
56	4598.257	4273.772
64	4623.147	4311.446
128	4672.584	4347.617
166	4579.127	4211.684
192	4719.967	4470.656
256	4632.422	4333.351
300	4724.322	4323.356

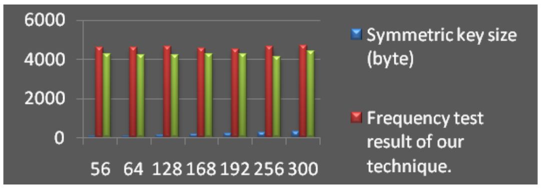


Figure 1. Graph of frequency test of Table 4

Observations: NIST SP 800-22 specifies that the randomness take a look at ought to follow the 3 traits inclusive of Uniformity, Scalability and Consistency. In case of uniformity and scalability, the prevalence of a 0 or 1 is equally in all likelihood this is the possibility of occurrence of 0 or one is half. The Table 4 of frequency check result indicates uniformity and scalability of our method.

In case of consistency, we can say that the seed data from which we will generate the session secret is symmetric key. For cryptographic applications, the symmetric key needs to be relaxed. The consultation key is generated by way of the usage of the idea of recurrence relation and DNA series. Now if the coefficient of recurrence relation is unknown or may exchange time to time and if the symmetric secret's secured then the following output bit within the sequence have to be unpredictable notwithstanding any understanding of preceding bits inside the collection.

It should no longer be viable to determine the symmetric key from the expertise of any generated values. There isn't any correlation among symmetric key and generated values. Thus our method proves the ahead and backward unpredictability. Furthermore, from the above desk and graph it's far visible that our proposed technique offers extra randomness than PRNG () that is wellknown technique.

Entropy test: Here we describe a comparative study between our technique and standard technique, PRNG () with session key and symmetric key. The following Table 5 and Figure 2 show the details of entropy value.

Table 5. Entropy value

Session key size (bits)	Entropy value of our technique.	Entropy value of PRNG()
56	6.88	7.00
64	6.81	7.01
128	6.82	7.02
166	6.84	7.03
192	6.87	7.01
256	6.89	7.05
300	6.90	7.06

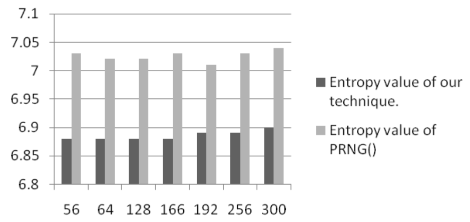


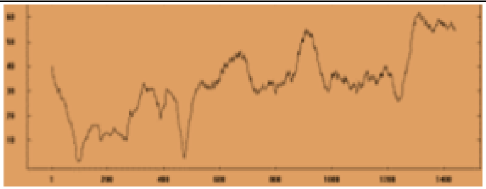
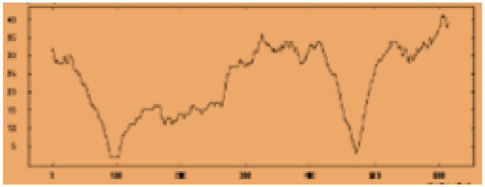
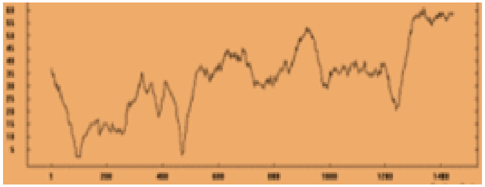
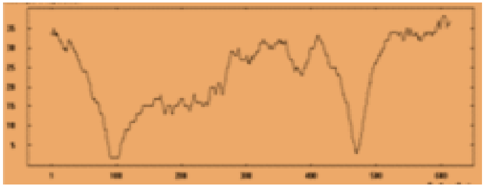
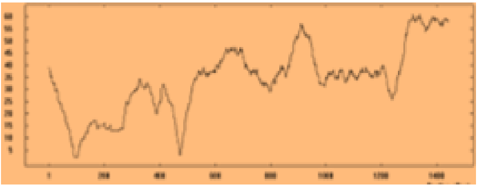
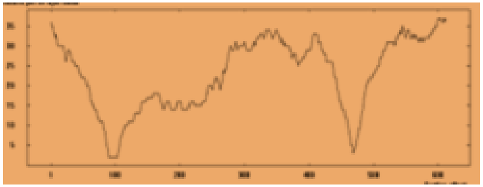
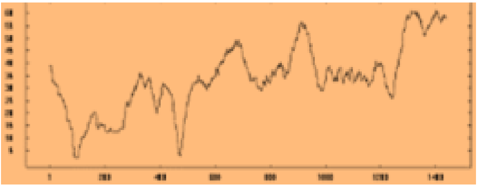
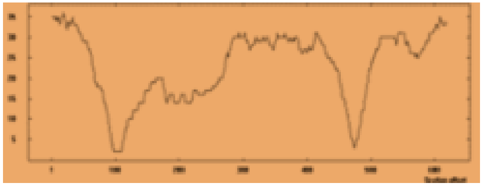
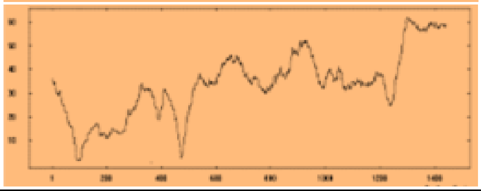
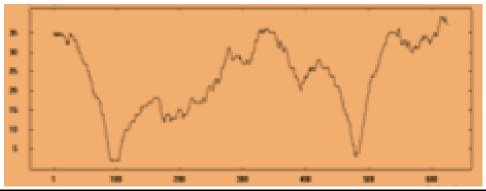
Figure 2. 2D graph of entropy value of Table 5

Observations: In cryptography, a cryptosystem is stated to be semantically comfortable if it is very difficult to an attacker to extract any information about the plain text from cipher text and its duration. Entropy can be described as randomness or unpredictability of information contained in a message. This randomness breaks the shape of undeniable textual content. Entropic security in encryption is just like semantic protection whilst records have pretty entropic distribution. Plain textual content entropy fee is zero. Now from the comparative has a look at of entropy price between our approach and PRNG (), it's far visible that the entropy fee of our approach is close to to the end result of PRNG (). The x-axis shows the key period. Thus, from the definition of entropic protection we say that it's miles impossible to are expecting simple textual content from cipher text if our approach is used to generate session key and the usage of this session key and symmetric key in encryption gives robustness.

6.3. Analsis of encryption technique

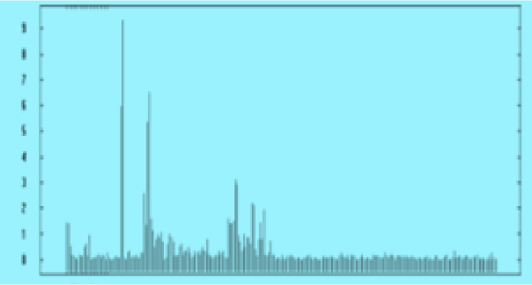
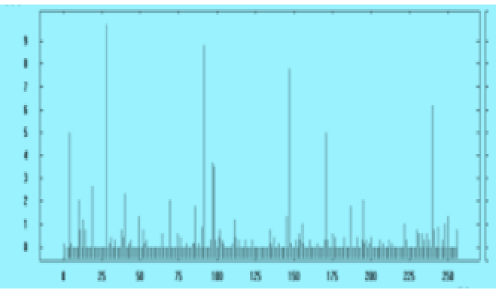
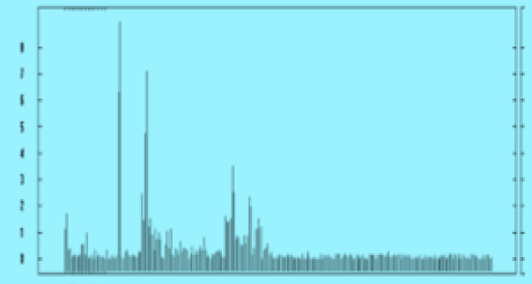
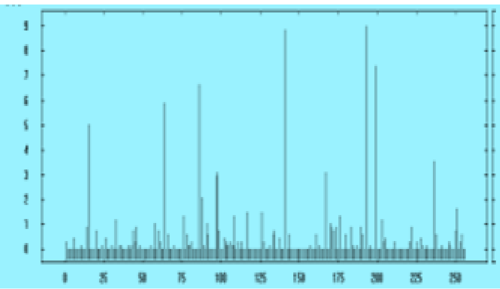
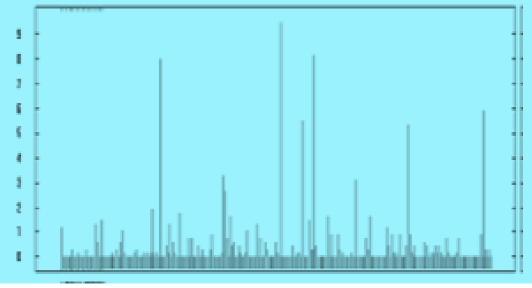
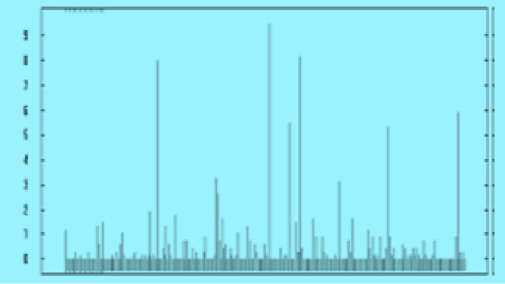
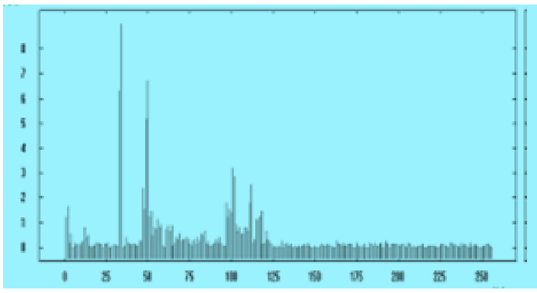
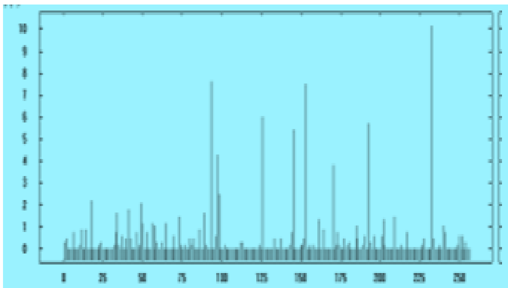
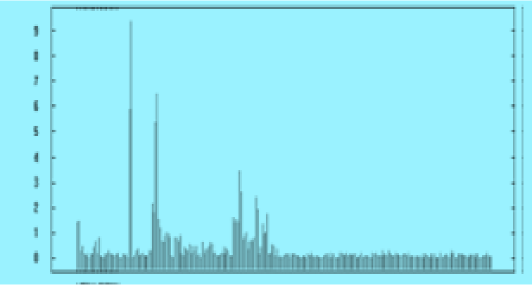
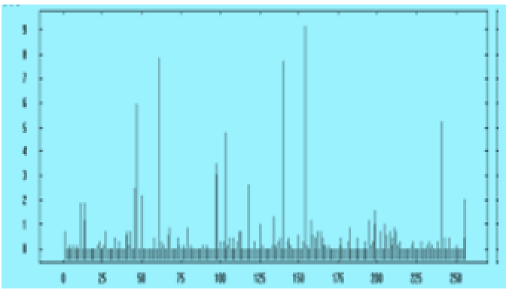
In section we have analysed the encryption technique through floating point frequency and histogram of each share [3, 22]. The following Table 6 shows the floating point analysis.

Table 6. Floating point analysis

Share	Plain text	Cipher text by proposed algorithm
Share 1		
Share 2		
Share 3		
Share 4		
Share 5		

Observation: The floating factor frequency analysis describes how binary values of a record are allotted along x-axis and y-axis. It is a graphical1 representation of a frequency distribution. Here we've tested the distribution of our information, such as the peaks, spread and symmetry of the cipher text and shared cipher text. The peaks represent the most not unusual values and unfold represents how a good deal our data varies. From the above Table 7, it has been discovered that the statistics aren't skewed. Histogram analysis:

Table7. Histogram analysis

Share	Plain Text	Cipher Text by Proposed algorithm
Sh 1		
Sh 2		
Sh 3		
Sh 4		
Sh 5		

Observation: The binary histogram describes in Table 8, how binary values of a file are distributed along x-axis and y-axis. The histogram analysis of the results shows that distribution is normal and normal in shape. The histogram of shares generated through the proposed methodology shows the data distribution in shared file are equal which proves the encryption using session key is good. Using any K number of shares we get back the encrypted file and from encrypted file it is infeasible to get an idea about session key. This proves strength of our scheme.

6.4. Comparative analysis

The following Table 8 shows the comparative analysis among secret sharing techniques and also proves the novelty of our scheme.

Table 8. Comparative analysis

Schemes Security Properties	Ref [16]	Ref [1]	Ref [22]	Ref [3]	Ref [4]	Proposed technique
Confidentiality	Yes	No	No	No	Yes	Yes
Integrity	No	No	No	No	Yes	Yes
Authenticity (message authentication and user authentication)	Yes	No	No	Yes	No	Yes
Privacy Protection	No	Yes	No	Yes	Yes	Yes
Defend against Man-in –middle attack	No	No	No	No	No	Yes
Vulnerability	No	No	No	Yes	Yes	Yes
Cryptanalysis (linear and differential)	No	No	No	Yes	Yes	Yes
Session key establishment	No	No	No	No	Yes	Yes

6.5. Authentication

For any cryposystem authentication is an important issue. In this article we have used two types of authentications one user authentication and other message authentication for our secret sharing. Two types of authentications are described below. User authentication [1, 6] ensures that only authorized users are gaining access to secret information or data. Without a secure authentication process, any transmission through network could be at risk. Certain login information are required for user authentication. Here we have used single factor authentication where every share is transmitted to the receiver end through its public key and each user takes its share using individual private key. These two keys are enough to comfirm the user’s identity and this will allow the system to authorize the user.

We have used MAC rules for message authentication, primarily based on symmetric key [1, 23, 24]. Here MD5 hash algorithm is used as MAC feature which compresses an arbitrary input period into a set duration output (128-bits). Sender transmits the message in conjunction with the MAC. After receiving the message and the MAC, the receiver recomputes the MAC value by using the symmetric key. If the computed MAC value does now not fit with the MAC which is despatched from the sender, receiver safely assumes that the message isn't the real.

6.6. Performance evaluation

In every test in result section, there is an observation part which provides the performance of our scheme on the basis of result analysis of the test. From the analysis of the observations, we say that our technique for secret sharing is better than previous techniques with respect to time complexity, implementation and robustness. In our paper we have introduced a new mask generation algorithm which is easy to implement and understand. This new mask generation process reduces the time complexity. So this is the better one than previous mask generation algorithm also The DNA sequence provides the nonlinearity in key generation. Random number generation from recurrence relation is also an important step in cryptography and different types of test and its results boosts the claim. Secret sharing is very relevant in big data and cloud computing environment. Our technique will be widely used in this environment for its simplicity, novelty and easy implementation.

7. CONCLUSION

Here we have presented a secured session key based secret sharing approach with minimum computation overhead. Here a new mask generation algorithm is introduced for secret share generation with low complexity. In this algorithm a unit matrix of specific order is taken. The number of participant is equal to order of matrix. The Session key is generated using the concept of recurrence relation and DNA sequence. Session key as well as secret data is shared among set of specific number of participants and these participants are able to reconstruct the original message. To the best of our knowledge this is the best threshold secret sharing approach, almost having minimum computational overhead for the duration of both proportion technology and reconstruction. Different types experimental results and analysis proves the efficiency and acceptability of our scheme.

REFERENCES

[1] Stallings William, *Cryptography and network security*, Pearson India Education Service Pvt. Ltd., 2015.

[2] C.Asmuth and J. Bloom, "A modular to key safeguarding," *IEEE Transaction on Information Theory*, vol.29, no. 2, pp. 208-210, 1983.

[3] G.R. Blakley, "Safeguarding cryptographic keys," in *979 International Workshop on Managing Requirements Knowledge (MARK)*, pp. 313, 1979.

[4] Y. Desmedt, "Some recent research aspects of threshold Cryptography," In *proceeding of ISW"97 1st International Information Security Workshop*, vol. 1196 of LNCS pp. 158-173 Springer-Verlag 1997.

[5] R. Guesmi, *et al*., "A novel chaos-based image encryption using DNA sequence operation and Secure Hash Algorithm SHA-2," *Nonlinear Dynamics*, vol. 83, no. 3, pp. 1123-1136, 2016.

[6] A. De Santis, Y. Desmedt, Y. Frankel and Y. Yung, "How to share a function securely?," In *proc of STOC 94*, pp. 522-533, 1994.

[7] Bhowmik A., Dey J., Sarkar A., Karforma S., "Computational intelligence based lossless regeneration (CILR) of blocked gingivitis intraoral image transportation," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 8, no. 3, pp:197-204, Sep. 2019.

[8] J.G. Chakravorty, P.R. Ghosh, *Advanced Higher Algebra*, U.N. Dhur and Sons Private Ltd., 2018.

[9] A. Peinado, J. Munilla, and A. F ´uster-Sabater, "EPCGen2 pseudorandom number generators: Analysis of J3Gen," Sensors, vol. 14, no. 4, pp. 6500-6515, 2014

[10] F. Zheng, X. Tian, J. Song, and X. Li, "Pseudo-random sequence generator based on the generalized henon map," *The Journal of China Universities of Posts and Telecommunications*, vol. 15, no. 3, pp.64–68, 2008.

[11] Sarkar A., Dey J., Chatterjee M., Bhowmik A., Karforma S., "Neural soft computing based secured transmission of intraoral gingivitis image in e-health care," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp. 178-184, Apr. 2019.

[12] Yunpeng Zhang *et al*., "An optimized DNA based encryption scheme with enforced secure key distribution," *Cluster Comput*, vol. 20, no. 4, pp. 3119–3130, 2017.

[13] Kari, L., Seki, S., Sosk, P., "DNA Computing—Foundations and Implications," *Handbook of Natural Computing*, Berlin: Springer, 2012.

[14] X. Wang, *et al*., "Image encryption algorithm based on multiple mixed hash functions and cyclic shift," *Optics and Lasers in Engineering*, vol. 107, pp. 370-379, 2018.

[15] A. Shamir "How to share a secret?," *Comm ACM*, vol. 22, no. 11, pp. 612-613, 1979.

[16] Prabir Kr. Naskar, Hari Narayan Khan, Ayan Chaudhuri, Atal Chaudhuri, "Ultra secured and authentic key distribution protocol using a novel secret sharing technique," *International Journal of Computer Applications*, vol. 19, no.7, pp. 12-15, Apr. 2011.

[17] Bhowmik A., *et al*., "Fuzzy-Based Session Key as Restorative Power of Symmetric Key Encryption for Secured Wireless Communication," *Proceedings of the 2nd International Conference on Communication, Devices and Computing*, Lecture Notes in Electrical Engineering vol. 602, Singapore: Springer, pp.171-184, 2020.

[18] A. Al Malki, M. M. Rizk, M. El-Shorbagy, and A. Mousa, "Hybrid genetic algorithm with K-Means for clustering Problems," *Open Journal of Optimization*, vol. 5, no. 02, p. 71, 2016.

[19] X. Chen, *et al*., "Adaptive control of multiple chaotic systems with unknown parameters in two different synchronization modes," *Adv. Differ. Equ.*, vol. 2016, pp. 1-17, 2016.

[20] Mandal, B. K., Bhattacharyya, D., & Bandyopadhyay, S. K., "Designing and performance analysis of a proposed symmetric cryptography algorithm," In *2013 International Conference on Communication Systems and Network Technologies*, pp. 453-461, 2013.

[21] Sivaperumal, S, "Hybrid synchronization of identical chaotic systems via novel sliding control with application to hyperchaotic vaidyanathan—volos system," *Int. J. Control Theory Appl.*, vol. 9, pp. 261–278, 2016.

[22] Preeti Singh *et al*., "Symmetric key cryptography: Current trends," *International Journal of Computer Science and Mobile Computing*, vol. 3 no. 12, pp. 410-415, 2014.

[23] H. F. Hua ng and C.C. Chang A "Novel efficient (t, n) threshold proxy signature scheme," *Information Sciences*, vol. 176, no. 10, pp. 1338-1349, 2006.

[24] Sarkar A., Dey J., Bhowmik A., "Multilayer neural network synchronized secured session key based encryption in wireless communication," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 1, pp169-177, Apr. 2019.

BIOGRAPHIES OF AUTHORS

Anirban Bhowmik completed Bachelor of Science (Mathematics) from Bolpur College, Bolpur, West Bengal, India and Master of Computer Application from the University of Burdwan in year 2008. He is working as an Assistant Professor in Department of Computer Application at Cyber Research & Training Institute, Burdwan, WB, India since 2008. He has published 17 conferences papers, journals and it’s also available online. His main research work focuses on Cryptography and Soft Computing. He has 12 years of teaching experience at UG level.

Sunil Karforma has completed his Bachelors in Computer Science & Engineering, and his Masters in Computer Science & Engineering, from Jadavpur University. He received his Ph.D. in Computer Science, and is presently Professor & Head of the Dept. of Computer Science at the University of Burdwan. His research interests include Network Security, E-Commerce, and Bioinformatics. He has published numerous papers in both national as well as international journals and conferences.

Joydeep Dey pursed Bachelor of Computer Application (Honours) from Cyber Research & Training Institute, Burdwan, India in 2007 and Master of Computer Application from the University of Burdwan in year 2011. He is working as Leturer in Department of Computer Sciences at M.U.C. Women’s College, Burdwan West Bengal, India since 2011. He has published two conferences papers and it’s also available online. His main research work focuses on Cryptography and Computational Intelligence.. He has 8 years and 0.5 years of teaching experience at UG and PG level respectively.