MIS Review Vol. 19, No. 1, September (2013), pp. 45-59
DOI: 10.6131/MISR.2013.1901.03

Securing DNA Information through Public Key Cryptography

Shiv P. N. Tripathi, Manas Jaiswal, Vrijendra Singh
Division of MS (Cyber Law & Information Security), Indian Institute of Information Technology, India

ABSTRACT: *This research work emerged as a new concept to provide robust security to the huge volume of information residing in DNA. In present scenario, security is being managed through symmetric key cryptography only. A new initiative has been taken to increase the robustness of DNA security. In this paper we are integrating public key cryptography inside traditional DNA security algorithm. The additional security is provided through a new algorithm as proposed, which takes advantage of residue theorem and traditional RSA algorithm. The main security concept is based on complexity in factorization and high versatility of choosing parameters/variables. Basically, DNA is encrypted through symmetric key cryptography and the key used to encrypt the data symmetrically is itself encrypted asymmetrically through proposed modified RSA algorithm. Through example, it is further illustrated in this paper that this is not only one of the optimized algorithms to provide a tradeoff between security and computational speed but also adds some sort of defense strategy against various attacks in a layered approach.*

KEYWORDS: *Information Security, Cryptography, DNA, DNA Security, Encryption, RSA Algorithm.*

1. Introduction

Information plays a vital role in today's generation and this information is very extensive and huge from the storage perspective. The computing methodology to get decisive result, DNA sequences must be used because it is the only source for "ultra-compact information storage" (Huge data stored in a compact volume). In a gram of DNA, 108 tera-bytes storage capacity is available (Gehani, LaBean & Reif, 1999) Total data present in the world can be stored in a few grams of DNA. Now a question arises "why there is a lag in adopting this technology for storage and transmission purpose?"

The answer lies in computational complexity and security of information transmitted. A lot of research has been done to increase the computational speed using vast parallelism (Cox, 2001; Cui, 2006). But the second aspect of DNA security is equally challenging and thirst area to be taken care of. The strands of DNA are used to encrypt the original data. In this technique data is using DNA. A literature survey has been done and it is found that generally either we are compromising with storage capacity, computational speed or security of data.

Now, here is a proposed hybrid security concept in which we combine the traditional approach with a new approach (DNA Cryptography). The extra advantage of this method is that it provides two levels of security (first is achieved by encrypting data symmetrically and second is through encrypting the key itself asymmetrically which is used to encrypt the data symmetrically). In this new algorithm we are using a new improved version of RSA that makes it too strong to break. It decreases its factorization possibility (the only way to crack RSA). The asymmetric key cryptography begins from a password / passphrase / key (that is used to encrypt data [achieved from the DNA sequence]) and this mechanism is introduced to enhance the security so that it could lead towards robustness in comparison to the traditional One Time Pad DNA, PCR, Index based symmetric encryption and asymmetric (digital signature method) (Cherian, Raj & Abraham, 2013).

2. Review of related literature

In a pioneering study, a lot of magnificent research has been done to encrypt and decrypt the information residing in DNA (Clelland, Risca & Bancroft, 1999; Leier, Richter, Banzhaf & Rauhe, 2000; Shimanovsky, Feng & Potkonjak, 2002; Youssef, Emam & Abd Elghany, 2012). A DNA sequence is formed with four of alphabets: A, T, G and C. Every alphabet among these four is a reference for a nucleotide. The two important factors for this sequence are shown:

(1) A real DNA is almost similar to a fake one.

(2) A large number of DNA sequences are openly available on the internet (EMBL-EBI, 2012). On the basis of referenced previous researches, we can get approximately 55 million DNA sequences openly available (EMBL-EBI; Youssef et al., 2012).

In this world, all the living creatures have their own unique genetic specification, which is formed by two strands of nucleotides, consisting one out of four alphabets as discussed above A, T, G, C. Apart from this formation of DNA, it has some sort of chemical polarization property, it clearly indicates that every molecule contains different chemical groups (Schena, 2003).

This was Adleman, with his outstanding work (Adleman, 1994), brought a revolution in this magnificent area of bio-computation research. This research of bio-computation helps to solve all the problems which were supposed either impractical or almost impossible to solve due to lack of tremendous amount of processing. By the use of exceptional and efficient DNA computation technology, it could have been possible to break Data Encryption Standard (DES) (Boneh, Dunworth & Lipton, 1995). The OTP cryptography was achieved through DNA strands and another milestone in this field was steganography, discussed in (Gehani et al., 1999).

Following the work of Gehani (Gehani et al., 1999), the pseudo cryptography was introduced by Ning Kang (Ning, 2009). In this technique, the original data is transformed into a DNA sequence. The resulting sequence is then transformed into two different forms of DNA. (1) Spliced form and (2) Protein form. To achieve this, introns are divided into specific patterns. In the above technique, instead of using an actual DNA sequence, DNA functions are used. Thus it is called Pseudo DNA cryptography. The basic idea of central dogma of molecular biology like Transcription, Translation and Splicing are used to replicate this technique.

With the rapid development of DNA cryptography, some biological and algebraic operations based on DNA sequence are presented by researchers (Mills, Yurke & Platzman, 1999; Soni & Johar, 2012; Wasiewicz, Mulawka, Rudnichi & Lesyng, 2000) such as addition or subtraction.

Using A, T, G and C strands of DNA several binary encryption techniques and java crypto encryption techniques are proposed (Leier et al., 2000; Tatiana, Mircea-Florin, Monica & Cosmin, 2008)

In another research, primers are employed as key for encrypting and decrypting data which will result in a DNA template (Cherian et al., 2013). The basic technology used in this scheme is Polymerase Chain Reaction (PCR) which is a DNA digital coding technique (Cui, Qin, Wang & Zhang, 2008). In this methodology input data is changed into hexadecimal values which will be converted further in to binary code. These binary digits are used to convert DNA sequence in to DNA template. PCR processes are executed (Clelland et al., 1999) by using the forward primer. Now that the DNA sequences has been changed through the above process, so it will be completely different from the original data. In the decryption process of achieved encoded data, we use reverse primer to get back original DNA sequences. We calculate the equivalent binary digits and transformed it into the original data.

Now, most of the DNA cryptography is done using only symmetric key schemes. An asymmetric encryption method was proposed by (Lai, Lu, Qin, Han & Fang, 2010). In this, two keys are used, one is for encryption and decryption and other is for creating signatures. LU MingXin using advanced cryptographic techniques proposed symmetric key crypto-system (Lu, Lai, Xiao & Qin, 2007), in which specially designed micro arrays were used and DNA fabrication and DNA hybridization is done for encryption and decryption.

3. Proposed methodology

DNA information algorithm proposes a hybrid cryptography technique (symmetric + asymmetric) (Terec, Vaida, Alboaie & Chiorean, 2011) to secure the data being transmitted via DNA. This uses the public as well as private key cryptographic methods to provide an additional level of security to encrypted data.

First, at the beginning, the sender and corresponding receiver will generate a pair of asymmetric keys. Second, a negotiation for the use of symmetric algorithm is done with the codon sequence (a particular sequence or arrangement in a logical order or a fixed pattern) to be used on the basis of DNA indexes. Third, there is an assumption in the message transfer that the data is encrypted with symmetric algorithm, while the key itself is encrypted asymmetrically and attached with the data. This type of double layered security approach was first given by (Nobelis, Boudaoud & Riveill, 2008). Given below is a detailed description of DNA algorithm (Generation, Encryption, Transmission and Decryption).

3.1 Generation of key pairs

In the beginning of generation phase, a password / password phrase (it could be complex enough based on the maturity of security level that has to be provided to the information residing in DNA). Basically the length and strength of password is totally dependent on the criticality of the data that has to be secured. After the initial hashing of the password and using improved RSA algorithm (discussed below) two keys are generated, the asymmetric key pairs generated will be different every time even if the user enters the same password as it is based on the randomness of pseudo prime numbers p & q.

3.2 Asymmetric key generation

In this section we are showing the methodology adopted (Wang, Chen & Duan, 2011) (improved version of RSA) for obtaining encoded key of DNA message. Basically this new algorithm adopts the basic idea of RSA algorithm but here the complexity is increased and made it very secure from the existing attack of RSA algorithm.

The main aim was to come up with an improved algorithm that will make us very comfortable from major attacks, key management, or key storage perspective.

Step 1: first the password / passphrase (to be chosen by the user) are converted into the byte array, and then its hash value is calculated.

Step 2: We transform the particular hash value into BIG integer number. This number shall be an odd number and is stored in a temp variable. Now this is the number that will basically undergo the given process.

Step 3: Choose two random, very large, prime numbers p & q such that $p < temp < q$.

Step 4: Choose any of the factor (it could be prime or composite), "f" of $\Phi(n) = (p - 1) \times (q - 1)$.

It will be very appropriate to choose any of the factors because in the traditional RSA we choose only prime factor and we are providing the flexibility and versatility to choose any of the factor. This will also make it unpredictable even in the easiest case of factorization of RSA.

Step 5: Choose an integer "e," such that (e, f) = 1, In other words we can say that e shall be relatively prime with f.

Step 6: Evaluate the value of integer "d," such that $d = e - 1 \pmod{f}$. Now we calculate the solution space s: $Xf \equiv 1 \pmod{n}$

Step 7: A subset s' of original solution space s in the above step is chosen at random, we can represent the subset s' as f' ($0 < f' < f$).

And s' is represented in terms of relational set as shown s' = {ai} (ai ε s && 1 <= i <= f). Now we do not need to reveal the value of f because if the value of f (the critical value for cryptanalysis) is not known, then it is impossible to judge the exact value for factorization in the big space of s'.

Step 8: The following keys that will be used are as follows:

- The triple (e, n, s') is considered as public key to encrypt the data.
- The (d, n, s') is considered as private key used for decryption purpose.

Here in the above discussed algorithm it is impractical to get the exact plain text with the help of cipher text. The main reason is that only cipher text and public key is known to the user and to break this, following condition shall satisfy:

A number X such that $X \epsilon s'$, (where ϵ indicates "belong to") so directly searching that particular number from the whole space of s' (i.e., is very large) is very much infeasible.

3.3 Encryption

The sender chooses three main factors of the algorithm:

- The symmetric algorithm to be used for encryption.
- The renewal period of the symmetric keys.

The renewal period or crypto period of the various keys are defined according to NIST standard document (Barker, Baker, Burr, Polk & Smid, 2011) Appendix. In this document it is defined in terms of SUP (Sender usage period) and RUP (Receiver usage period) as reflected in Appendix. In purview of both time periods, largest one is chosen as renewal period or crypto period. The list of various time periods for different time period is given in Appendix.

- The length of the symmetric key.

Now codon sequences are chosen as it will specify where index search is to be done. The negotiation begins when the sender transfers its public key to the corresponding receiver. The public key of receiver is encrypted with the public key of the sender and retransmitted to the sender. The sender will encrypt the chosen parameters with the receiver's public key which he received. The receiver on receiving the parameter might reject them on a factor of providing maximum security. In this case, a renegotiation is done until both are convinced on some security parameters.

The whole encryption process is clearly indicated in the Figure 1. From the programming perspective (used in this research) as shown in the below drawn box, provider public class is providing the basic medium for storing the data in DNA itself. Now various cryptographic techniques have been used to provide the security to this storage class of DNA information. The basic skeleton of programming of encryption and decryption process is as follows:

The basic implementation of a DNA security is as follows:

```
public class DNASecurity extends Provider
{
public DNASecurity
{
super ("DNAProvider", "DNASecurity");
}
}
```

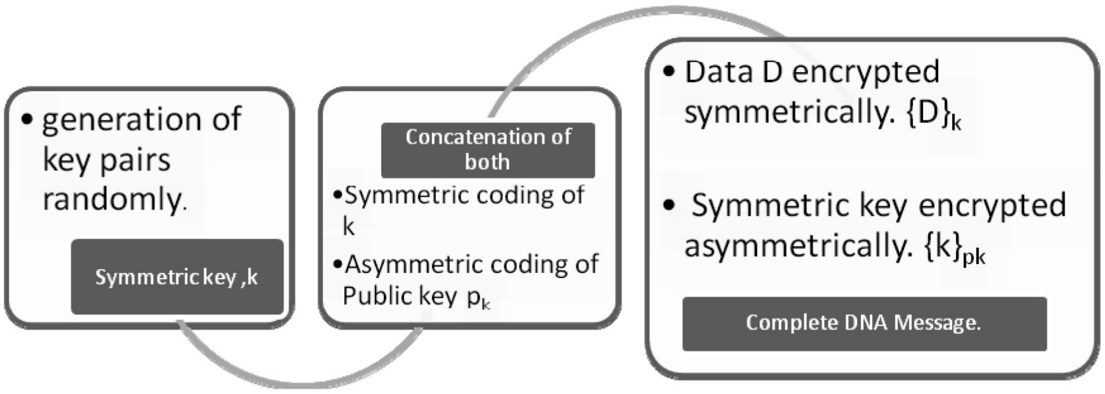

• generation of key pairs randomly.
Symmetric key ,k
Concatenation of both
•Symmetric coding of k
•Asymmetric coding of Public key p_k
• Data D encrypted symmetrically. $\{D\}_k$
• Symmetric key encrypted asymmetrically. $\{k\}_{pk}$
Complete DNA Message.

Figure 1 Complete DNA Encryption Process

3.4 Transmission

The encryption phase ends with the transmission of test message, if the transmitted message experiences an error or if too much delay in transmission is present, then the encryption phase needs to be evaluated again. Before the transmission of data can begin, the actual data is encoded using symmetric key, according to the negotiations made earlier. At regular time period "t," symmetric key is generated and receiver's public key is used to encrypt the symmetric key and integrate with original message that has to be transmitted. Henceforth data is encrypted with symmetric key, which is in turn encrypted with receiver's public key. The efficiency and robustness of this approach is even greater than full-fledged algorithm, this is due to the fact that symmetric key is itself encrypted with asymmetric key (through improved version of RSA algorithm)

Now data shall be transformed to a byte array, further it is transformed to raw DNA message. This is done by substitution cipher cryptographic technique by substituting the alphabets. The obtained message is then transformed into indexes i.e., transformed in string form (Figure 1).

3.5 Decryption

The decryption process is much similar to encryption process. The indexes generated in transmission phase are converted into raw DNA array. Now extraction of the data is performed through symmetric key generated by the private key of the receiver and then all negotiations and its data are destroyed.

```
import java.io.*;
import java.lang.String;
      class decryption {
      public static void main (String args []) throws Exception
                                              {
                                                        *****Main Body*******
                                              }
                        }
```

Where, D is denoted for data that has to be encrypted, K is denoted as key based on chosen algorithm & P_k is denoted as receiver's public key.

Now according to the Figure 1, complete DNA message is pretty much wrapped with two layers of security. In the first block normally DNA message is encrypted with any of the symmetric key cryptography method then the upper layer security is provided by encrypting the key of symmetric key through asymmetric key cryptography as shown in the second block. Third block shows the complete package of DNA message that has to be transmitted over the insecure network. Thus we are showing in the proposed methodology that we are not only providing the extra layer of security but also providing less complexity (from sender perspective) and high speed (as illustrated in the further section) without compromising the security.

4. Simulation result and analysis

This work clearly indicates, the obtained result stated by previous research work and new work with proposed RSA algorithm (Figure 2) and its comparison with improved version of RSA and other symmetric algorithm (Figure 3). Here we just tested DES, Triple DES, AES, IDEA and improved version of RSA algorithm. The motivation behind this is just to do comparative study of various cryptographic algorithms in terms of time complexity.

The test message (Figure 4) shown on the console input is used for testing the time complexity. The results shown below (Figure 2 & 3) are checked over stated configuration (Table 1). Where, time shown on Y axis is measured in milliseconds.

In the stated results (Figure 2 & 3), we are seeing that an exception is being generated i.e., Encryption and Decryption time in DNA cryptography algorithm is on a little bit higher side but we encounter this problem due to java platform used for simulation. Actually in java, cryptography algorithm is implemented in two parts. First one is public class that could be used by instance of these classes, second is an abstract

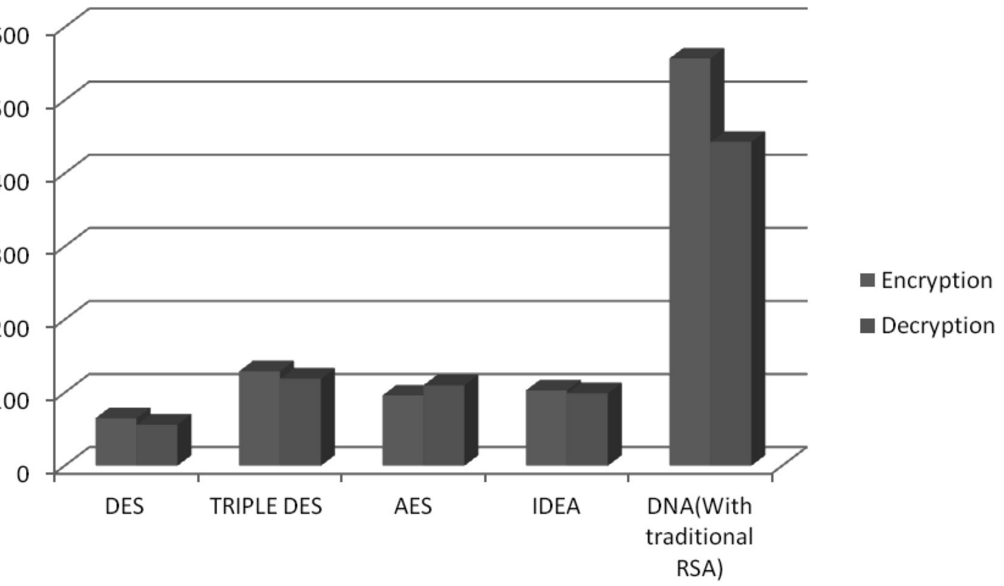


Figure 2 Time Complexity Comparison with Various Cryptographic Algorithms

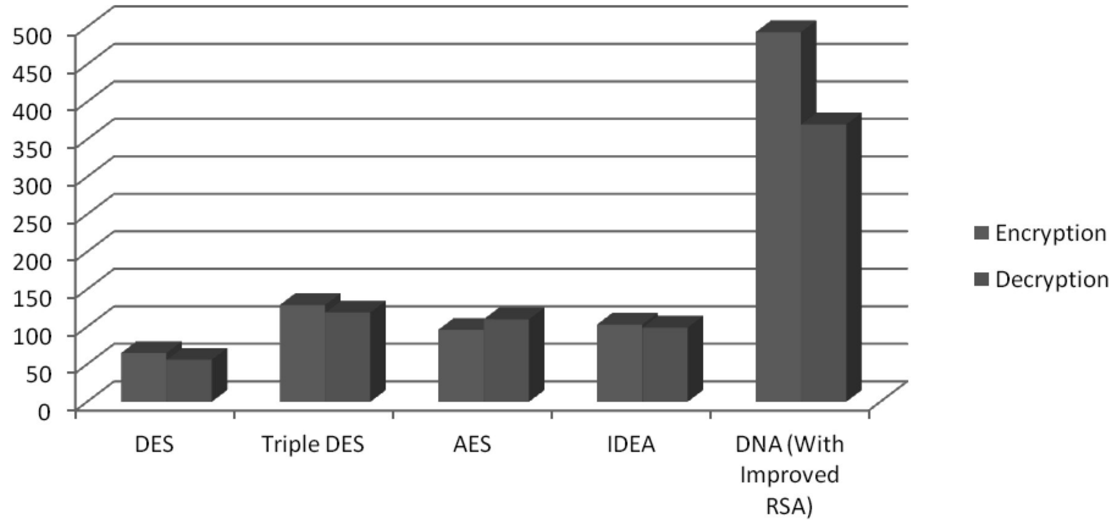


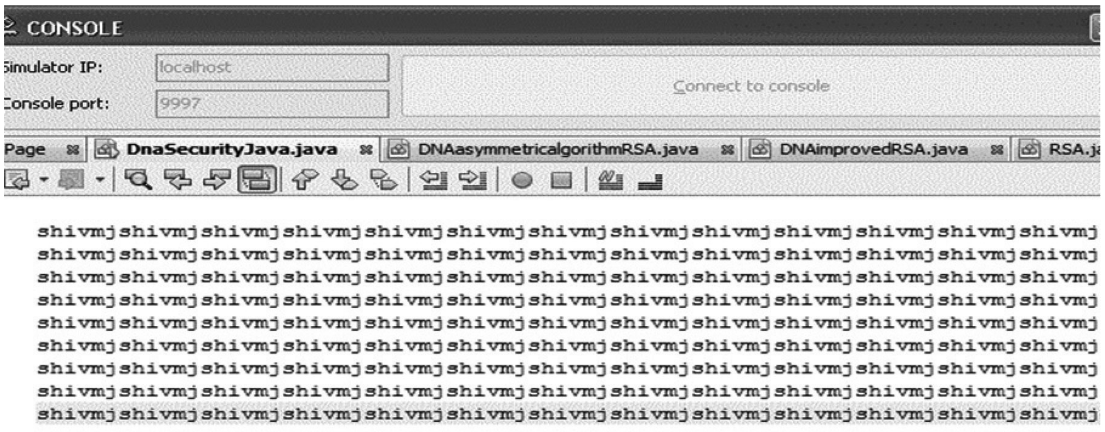
Figure 3 Time Complexity Comparison of Improved RSA (Proposed) with Various Cryptographic Algorithms

class; one has to inherit this class that simply needs some extra time. Now, the one more aspect of this simulation result is total computation time that has been calculated through below entered sequences that are shown on the console input (Figure 5).

It is very important to know for all of us that the total computation time is completely dependent on the OS platform and situation/environment in which we are simulating our

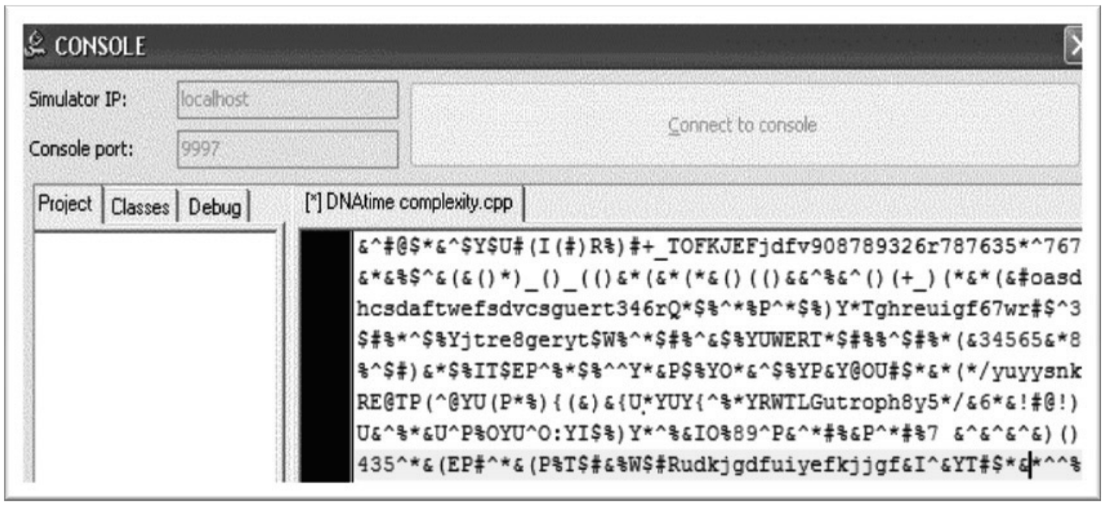
Table 1 Configuration Used for Testing Time Complexity

Processor	Specification	RAM	Operating System	OS Specification
Intel® Core™ 2 Duo	T5800, 2.00 GHz	1 GB	Windows 8	32 bit



```
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
shivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmjshivmj
```

Figure 4 Console Input for Testing Time Complexity



```
&^#@$*&^$Y$U#(I(#)R%)#+_TOFKJEFjdfv908789326r787635*^767
&*&%$^&(&()*)_()_(()&*(&*(*&()(()&&^%&^()(+_)(*&*(&#oasd
hcsdaftwefsdvcsguert346rQ*$%^*%P^*$%)Y*Tghreuigf67wr#$^3
$#%*^$%Yjtre8geryt$W%^*$#%^&$%YUWERT*$#%%^$#%*(&34565&*8
%^$#)&*$%IT$EP^%*$%^^Y*&P$%YO*&^$%YP&Y@OU#$*&*(*/yuyysnk
RE@TP(^@YU(P*%){(&)&{U*YUY{^%*YRWTLGutroph8y5*/&6*&!#@!)
U&^%*&U^P%OYU^O:YI$%)Y*^%&IO%89^P&^*#%&P^*#%7 &^&^&^&)()
435^*&(EP#^*&(P%T$#&%W$#Rudkjgdfuiyefkjjgf&I^&YT#$*&*^^%
```

Figure 5 Console Input for Testing Computation Time

results. This could vary according to situation and how this platform is installed. If this is calculated over normal installed platform (Table 2) then we get these results (Table 3).

As shown in the above Table 3, the total computation time of the DNA security algorithm with this hybrid approach (Proposed Algorithm) is very much larger than in perfect case of symmetric key cryptography but still less than with traditional asymmetric algorithm (RSA). So this is the whole idea behind DNA information security through

Table 2 Configurations Used for Testing Computation Time

Serial No.	Processor	Specification	RAM	Operating System	Specification
Configuration 1	Intel Dual Core	E5200 @2.50 GHz	2 GB RAM	Windows XP SP 2	32 Bit
Configuration 2	Intel Core i3	M350 @ 2.27 GHz	1 GB RAM	Ubuntu 10.04	Linux

Table 3 Result Obtained for Computation Time for Different Configuration

Algorithm	Configuration 1		Configuration 2	
	Encryption	Decryption	Encryption	Decryption
DES*	1.3	0.77	0.9	0.2
	1.32	0.65	0.62	0.23
	1.36	0.93	1.02	0.46
Triple DES*	4.1	1.12	3.2	0.79
	4.9	1.62	3.9	0.77
	5.2	3.2	3.8	0.72
IDEA*	10	9.3	9.1	5.2
	10.6	7.2	10.4	6.0
	11.2	6.3	11.0	5.8
AES*	1.3	0.13	1.0	0.10
	0.6	0.15	0.5	0.32
	2.3	0.89	2.1	0.93
DNA Security through RSA*	556	6.29	551	2.00
	553	5.2	493	2.62
	439	4.19	425.3	2.32
DNA Security through Improved RSA* (Proposed)	493	4.00	446.7	2.65
	452	3.91	449	3.2
	392.6	3.9	452.1	2.93

Note: *denotes the time taken for encryption and decryption process in "milliseconds."

public key cryptography consisting of double layers i.e., defense in depth security. DNA encryption and decryption itself is a very complex and time taking process and in this case of implementation (discussed in this paper) object oriented language & JDK platform is used in which abstract classes have been inherited that increase approximately 15% ~ 20% time in encryption and decryption time. Average total computation time in DNA security (proposed algorithm) is 445.866 milliseconds, and this is expected that if this would be implemented through parallel processing then definitely we shall get computation time reduced.

5. Conclusion and future scope

In this era of information technology, we need a huge and secure space to store our information. As it is discussed in the paper that in a gram of DNA 108 tera byte information is stored and through DNA cryptography we have proposed a very technical, efficient and effective way to secure the information residing in DNA. Although it has been shown that DNA Cryptography is itself a time consuming process and encrypting asymmetrically takes more time(as shown in the results). In the above discussion we proposed a completely new way to encrypt the information contained in DNA by integrating proposed new asymmetric RSA algorithm. This approach provides higher stability (it takes more time to factorize & cryptanalysis) and more powerful (it provides double layered encryption i.e., impractical to break) than any other symmetric algorithm like INDEX based, OTP, DES, and AES. Apart from security concept, it is very clearly demonstrated that windows is more time consuming in terms of computation time in comparison to Linux particularly for the algorithm test. Ultimately in a practical environment we have shown the direct way to manage and secure the information. The problem of excess computation time and less speed can be improved with the help of parallel processing. In future, higher speed can be achieved through parallel processing.

References

Adleman, L.M. (1994), 'Molecular computation of solutions to combinatorial problems', *Science*, Vol. 266, No. 5187, pp. 1021-1024.

Barker, E., Baker, W., Burr, W., Polk, W. and Smid, M. (2012), 'Recommendation for key management -- part 1: general', NIST Special Publication 800-57, National Institute of Standards and Technology, Gaithersburg, MD.

Boneh, D., Dunworth, C. and Lipton, R.J. (1995), 'Breaking DES using a molecular computer', *Proceedings of DIMACS Workshop on DNA Based Computers*, Princeton, NJ, pp. 37-66.

Cherian, A., Raj, S.R. and Abraham, A. (2013), 'A Survey on different DNA cryptographic methods', *International Journal of Science and Research*, Vol. 2, No. 4., pp. 167-169.

Clelland, C.T., Risca, V. and Bancroft, C. (1999), 'Hiding messages in DNA microdots', *Nature*, Vol. 399, pp. 533-534.

Cox, J.P.L. (2001), 'Long-term data storage in DNA', *Trends in Biotechnology*, Vol. 19, No. 7, pp. 247-250.

Cui, G.Z. (2006), 'New direction of data storage: DNA molecular storage technology', *Computer Engineering and Applications*, Vol. 42, pp. 29-32.

Cui, G.Z., Qin, L., Wang, Y. and Zhang, X. (2008), 'An encryption scheme using DNA technology', *Proceedings of IEEE 3rd International Conference on Bio-Inspired Computing: Theories and Applications*, Adelaide, SA, pp. 37-42.

EMBL-EBI. (2012), 'The European Bioinformatics Institute, Part of the European Molecular Biology Laboratory', available at http://www.ebi.ac.uk (accessed 20 December 2013).

Gehani, A., LaBean, T.H. and Reif, J.H. (1999), 'DNA-based cryptography', *Proceedings of 5th Annual DIMACS Meeting on DNA Based Computers*, Cambridge, MA, pp. 233-249.

Lai, X.J., Lu, M.X., Qin, L., Han, J.S. and Fang, X.W. (2010), 'Asymmetric encryption and signature method with DNA technology', *Science China Information Sciences*, Vol. 53, No. 3, pp. 506-514.

Leier, A., Richter, C., Banzhaf, W. and Rauhe, H. (2000), 'Cryptography with DNA binary strands', *BioSystems*, Vol. 57, No. 1, pp. 13-22.

Lu, M.X., Lai, X.J., Xiao, G.Z. and Qin, L. (2007), 'Symmetric-Key cryptosystem with DNA technology', *Science China*, Vol. 50, No. 3, pp. 324-333.

Mills, A.P. Jr., Yurke, B. and Platzman, P.M. (1999), 'Article for analog vector algebra computation', *BioSystems*, Vol. 52, No. 1-3, pp. 175-180.

Ning, K. (2009), 'A pseudo DNA cryptography method', available at http://arxiv.org/abs/0903.2693 (accessed 25 December 2013).

Nobelis, N., Boudaoud, K. and Riveill, M. (2008), 'Une architecture pour le transfert électronique sécurisé de document', Unpublished PhD dissertation, Equipe Rainbow, Laboratories I3S-CNRS, Sophia-Antipolis, France.

Schena, M. (2003), *Microarray Analysis*, John Wiley & Sons, Hobooken, NJ.

Shimanovsky, B., Feng, J. and Potkonjak, M. (2002), 'Hiding data in DNA', *Proceedings of 5th International Workshop on Information Hiding*, Noordwijkerhout, The Netherlands, pp. 373-386.

Soni, R. and Johar, A. (2012), 'An encryption algorithm for image based on DNA sequence addition operation', *World Journal of Science and Technology*, Vol. 2, No. 3, pp. 67-69.

Tatiana, H., Mircea-Florin, V., Monica, B. and Cosmin, S. (2008), 'A java crypto implementation of DNAProvider featuring complexity in theory and practice', *Proceedings of 30th International Conference on Information Technology Interfaces*, Dubrovnik, Croatia, pp. 607-612.

Terec, R., Vaida, M.F., Alboaie, L. and Chiorean, L. (2011), "DNA security using symmetric and asymmetric cryptography', *International Journal on New Computer Architectures and Their Applications*, Vol. 1, No. 1, pp. 34-51.

Wang, R., Chen, J. and Duan, G. (2011), 'A k-RSA algorithm', *Proceedings of IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, China, pp. 21-24.

Wasiewicz, P., Mulawka, J.J., Rudnichi, W.R. and Lesyng, B. (2000), 'Adding numbers with DNA', *Proceedings of 2000 IEEE International Conference on Systems, Man and Cybernetics*, Nashville, TN, pp. 265-270.

Youssef, I.M., Emam, A. and Abd Elghany, M. (2012), 'Multi-layer data encryption using residue number system in DNA sequence', *International Journal of Security and Its Applications*, Vol. 6, No. 4, pp. 1-12.

About the authors

Shiv P. N. Tripathi is pursuing MS in information Security from Indian Institute of Information Technology, Allahabad (UP) India. In the past, he has completed his B.Tech (Hons.) in Information Technology from BBD National Institute of Technology and Management in 2012. He has worked (as a co-author) in a research paper titled "Basics of an Affordable & Ubiquitous 5G Wireless Network." He has been awarded as an excellent child scientist (as a project leader) in 2006 in National child science congress.
Corresponding author. Division of MS (Cyber Law & Information Security), Indian Institute of Information Technology, Allahabad, Uttar Pradesh, Pin: 211012, India. Tel: +91-9532890146. E-mail address: tripathi161290@gmail.com

Manas Jaiswal is pursuing MS in information Security from Indian Institute of Information Technology, Allahabad (UP) India. In the past, He has completed his B.Tech in Computer Science from Amity University in 2011. He is member of Microsoft Tech. Club sponsored by Microsoft student partner India. E-mail address: manasjswl@gmail.com

Vrijendra Singh is an Associate Professor and a member of academic council in Indian Institute of Information technology, Allahabad (UP), India. He has been a member of various professional and research societies like IEEE, International Association of Engineers. He has 10+ years of experience in research and teaching field. He has published more than 20 papers in various international or national publications. His major area of interests are Artificial Neural Networks, Information security, Image processing, Cryptography etc. E-mail address: Vrijendra.singh@gmail.com

Appendix

In the NIST document recommended crypto period for specific key types has been given (Barker et al., 2012).

Key Type	SUP	RUP
Private Signature Key		1 ~ 3years
Public Signature Key		Several years (depends on key size)
Symmetric Authentication Key	≤ 2years	≤ SUP + 3years
Private Authentication Key		1 ~ 2years
Public Authentication Key		1 ~ 2years
Symmetric Data Encryption Key	≤ 2years	≤ SUP + 3years
Symmetric Key Wrapping Key	≤ 2years	≤ SUP + 3years
Symmetric and asymmetric RNG Keys		Upon reseeding
Symmetric Master Key		About 1year
Private Key Transport Key		≤ 2years
Public Master Key		1 ~ 2years
Symmetric Key Agreement Key		1 ~ 2years
Private Static Key Agreement Key		1 ~ 2years
Public Static Key Agreement Key		1 ~ 2years
Private Ephemeral Key Agreement Key		One key agreement transaction
Public Ephemeral Key Agreement Key		One key agreement transaction
Symmetric Authorization Key		≤ 2years
Private Authorization Key		≤ 2years
Public Authorization Key		≤ 2years

Source: Barker et al. (2012).

Where, Key denotes the type of key what we are using in cryptographic transaction. SUP denotes the sender's usage period. RUP denotes the receiver's usage period.