DNA based Cryptography: an Approach to Secure Mobile Networks

Harneet Singh
Computer Science & Engineering Deptt
Thapar University, Patiala

Karan Chugh
Computer Science & Engineering Deptt
Thapar University, Patiala

Harsh Dhaka
Computer Science & Engineering Deptt
Thapar University, Patiala

A. K. Verma
Computer Science & Engineering Deptt
Thapar University, Patiala

ABSTRACT

Security has always been the main concern in data communication and networking. Mobile Networks are highly vulnerable to security attacks and pose a great challenge for the wireless networks being used today. Since the mode of communication is open, these networks do not carry any inherent security and hence are prone to attacks. The present day algorithms have shown limitations to meet the security requirements of transmission. DNA cryptography is a new and promising direction in the field of cryptography. This paper proposes DNA-based Cryptography as an approach to ensure highly secure environment for transmission of data across mobile networks.

Categories and Subject Descriptors

D.4.6 [Security and Protection]: Cryptographic Controls

General Terms

Algorithms and Security

Keywords

Mobile Networks, DNA Cryptography, Security

1. INTRODUCTION

The era of Personal Computers is changing to an era of Ubiquitous Computing. Wireless networks have succeeded as they provide a better solution for interconnection of ubiquitous devices [1][7]. Mobile networks, characterized by anytime, anywhere communication [9]; the next generation of wireless communication systems, are an autonomous system of mobile routers and associated hosts that are connected by wireless links. The approaches applied in wired networks cannot be used for mobile networks due to vast differences in the characteristics of both the networks in terms of cost, power consumption and computational abilities [8][5]. Since the communication takes place in the open air, lack of centralized monitoring and management point, these networks are prone to attacks. Security in a network based on cryptography provides several aspects such as confidentiality, integrity, authenticity and non repudiation [3].

DNA cryptography is based on central dogmas of molecular biology [1]. However, pseudo DNA cryptography is different from actual DNA cryptography. The proposed method does not use biological DNA sequences (or oligos) or the sequences generated in-vitro, but only the DNA terminology and mechanisms of DNA function [2][6]. The cipher and decipher processes are based on the concepts of DNA transcription, splicing and RNA translation [2].

The structure of the paper is organized as: Section 2 reviews the related concepts, Section 3 describes the proposed methodology of using DNA based Cryptography. Section 4 discusses analysis drawn from the findings and Section 5 concludes the paper.

2. RELATED WORK

Mobile Networks are wireless, open, temporarily meshed networks composed of a group of mobile nodes. Each node acts as a router and forwards packets to other nodes to reach destination [8]. As no fixed infrastructure is required for their establishment, they are highly self-organizing. Mobile networks are characterized by feature of having distributed approach, dynamic topography and peer to peer analogies. Various proactive routing protocols like Open Shortest Path First (OSPF), Destination Sequenced Distance Vector (DSDV) and Reactive i.e. on-demand protocols like Ad hoc On-demand Distance Vector (AODV) and Hybrid routing protocols like Zone Routing Protocol (ZRP) are available which assume collaboration between nodes so they lack any embedded security mechanism and hence are more prone to security attacks [5]. These attacks can either be active attacks or passive attacks. Active attacks harm the network resources such as denial of service and modify the information being transferred. On the contrary, passive attacks without harming the network resources acquire the information and use it for unauthorized purposes such as releasing the message contents [4].

Modern day cryptography includes the process of encryption and decryption along with the involvement of various distinct mechanisms such as symmetric or asymmetric key encipherment and hashing [4]. In symmetric cryptography, both the encryption and decryption keys are same and need to be exchanged between the sender and receiver beforehand. Asymmetric cryptosystems use different keys for encryption and decryption; the encryption key is public and decryption key is retained by its owner. The proposed cryptographic algorithm follows symmetric cryptographic scheme.

3. PROPOSED ALGORITHM

The following pseudo code provides an insight into the methodology used. The sender node converts the original message into cipher text using the following steps:

```
BEGIN
Step 1: Select the message, M, to be sent, and convert into an 8
        bit Extended ASCII code, $M_{bin}$.
Step 2: Convert $M_{bin}$ into DNA notation, say $M_{dna}$ using the
        following convention: A=00, T=01, G=10, C=11 where
        A, T, G, C are DNA base pairs. The $M_{dna}$, as per
        analogy, comprises of exons and introns.
Step 3: Select the pattern to be spliced (introns), say S from
        $\Phi(M_{dna})$ where $\Phi(M_{dna})$ is the function that determines
        the random pattern to be spliced. The message becomes
        $M*_{dna}$ such that $M^*_{dna}= M_{dna} – nS$, where n is the
        number of the times the pattern appears in the $M_{dna}$.
        // $M^*_{dna}$ comprises of exons only and forms m-RNA
           sequence which is used for protein synthesis.
Step 4: The positions from where the pattern is spliced, the
        spliced pattern and the position of splicing are added to
        the key file, K1.
Step 5: Compute the length of $M^*_{dna}$, say $l(M^*_{dna})$.
Step 6: Set Flag= $l(M^*_{dna})$ mod 3.
        CASE I: If flag= 0 then
          i.   Convert $M^*_{dna}$ into amino acid sequence $M_{amino}$ using
               $\Theta(M^*_{dna})$ that combine 3 bases (codon) in $M^*_{dna}$ to
               form an equivalent amino acid using genetic code
               table.
        CASE II: If flag= 1 then
          i.   Compute the complementary base to the last base in
               the $M^*_{dna}$ using $C(M^*_{dna})$ where $C(M^*_{dna})$ computes
               the complementary base to the last base in $M^*_{dna}$ .
          ii.  Append the complementary bases at the end of
               $M^*_{dna}$ twice. Let new message be $M^*_{dna}= M^*_{dna} +
               C(M^*_{dna}) + C(M^*_{dna})$ .
          iii. Convert $M^*_{dna}$ into amino acid sequence $M_{amino}$
               using $\Theta(M^*_{dna})$ that combines 3 bases (codon) in
               $M^*_{dna}$ to form an equivalent amino acid using
               genetic code table.
        CASE III: If flag= 2 then
          i.   Compute the complementary base to the last base in
               the $M^*_{dna}$ using $C(M^*_{dna})$ where $C(M^*_{dna})$ computes
               the complementary base to the last base in $M^*_{dna}$..
          ii.  Append the complementary base at the end of $M^*_{dna}$
               once. Let new message be $M^*_{dna}= M^*_{dna} + C(M^*_{dna})$.
          iii. Convert $M^*_{dna}$ into amino acid sequence $M_{amino}$ using
               $\Theta(M^*_{dna})$ that combines 3 bases (codon) in $M^*_{dna}$ to
               form an equivalent amino acid using genetic code
               table.
Step 7: The mapping details from codon to amino acid and the
        flag value are added to the key file, K2.
END
```

The above methodology can be summarized as shown in Figure 1 below:

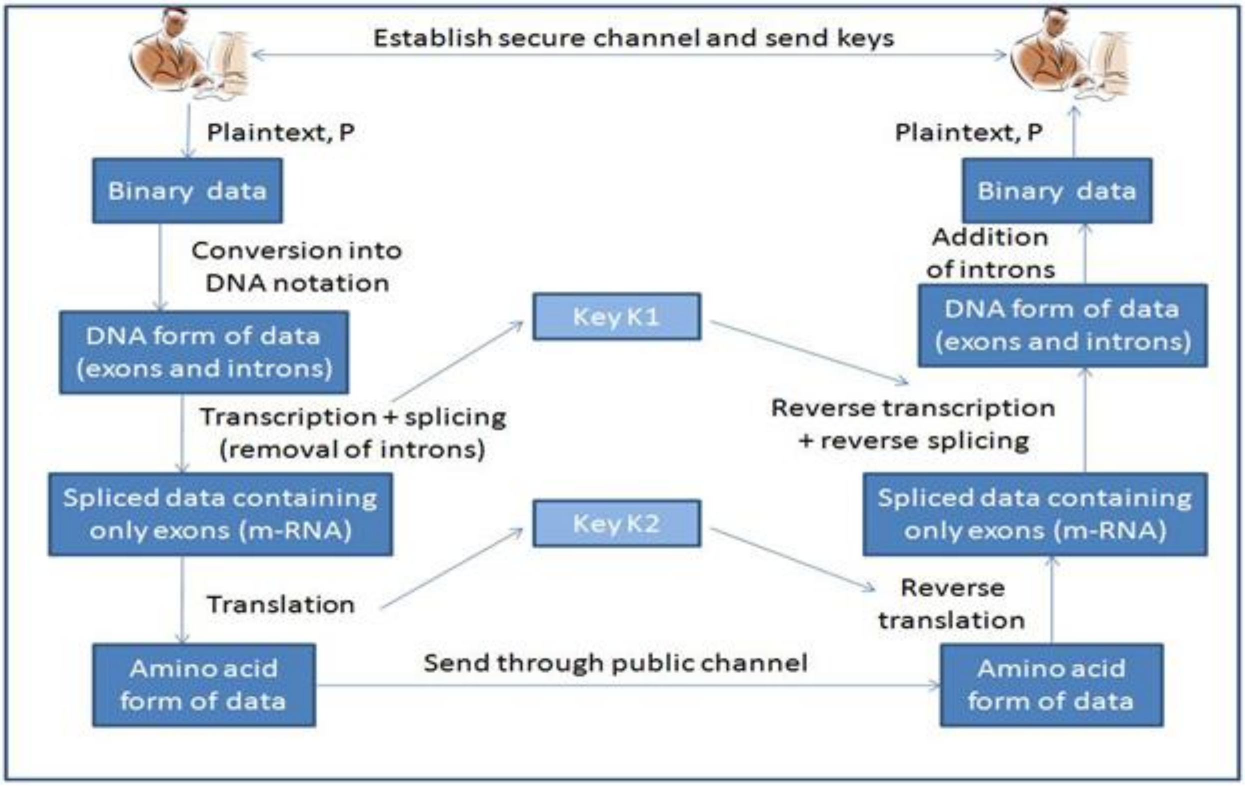
Establish secure channel and send keys
Plaintext, P
Binary data
Conversion into DNA notation
DNA form of data (exons and introns)
Key K1
Transcription + splicing (removal of introns)
Spliced data containing only exons (m-RNA)
Key K2
Translation
Amino acid form of data
Send through public channel
Amino acid form of data
Reverse translation
Spliced data containing only exons (m-RNA)
Reverse transcription + reverse splicing
DNA form of data (exons and introns)
Addition of introns
Binary data
Plaintext, P

Figure 1: The Communication Process

The receiver obtains the original message from the cipher text and keys using the following procedure:

BEGIN

Step 1: The message M_{amino} is converted into M^*_{dna} using the (Θ_R, K2) where Θ_R is the reverse of Θ such that $M^*_{dna} = \Theta_R(M_{amino})$.

Step 2: Using the value of Flag cut the appended bases from M^*_{dna} .

Step 3: M^*_{dna} comprises only of exons and process of Reverse splicing (Φ_R, K1) such that $M_{dna} = \Phi_R(M^*_{dna})$.

Step 4: The message is converted into binary, M_{bin} form from DNA notation.

Step 5: M_{bin} is in Extended ASCII with respect to original message which is converted back using reverse convention.

END

4. FINDINGS AND ANALYSIS

Suppose the DNA form of data M_{dna} have the length '*m*'. Let there be '*i*' introns and the average length of introns be '*l*'. So the length of the data after the introns are spliced from the DNA would be *m-i*l*. Since one codon consists of 3 bases so the length of the protein form of data would be *(m-i*l)/3*.

It is found that the sender needs to traverse the complete data once for splicing the introns from the DNA. So the time complexity of the splicing process is *O(m)*. For translation, the M^*_{dna} is traversed only once leading to complexity *O(m)*. Hence, the total time complexity of the encryption process is *O(m)*. At the receiving end, the cipher text is traversed once each for both the keys to obtain the plaintext in linear time with a total time complexity of *O(m)*.

It is analyzed that if some malicious node captures the data during the transfer between the nodes, it can only get cipher text. The probability of obtaining plaintext from the cipher text is very low even if brute force method is applied. To obtain M^*_{dna} from M_{amino}, 20 amino acids are to be mapped to 61 codons, thereby leading to 3 possibilities for every amino acid on an average. So, there would be $3^{(m-i*l)/3}$ total possible combinations to obtain the correct M^*_{dna}. Now to obtain M_{dna} there are *(m-i*l)+1* possible places for the insertion of intron. Every time an intron is inserted, the number of possible places for the insertion of intron also increases by 1. Since there are *i* introns, so the total combinations for reverse splicing are *(i*(2(m-i*l)+i+1)/2)*, which is of the order *O(m)*. As the number of introns and their length decreases, the time complexity of reverse splicing will decrease but the time complexity of reverse translation will increase. Hence, the total possible combinations for the decryption using brute force are $(3^{(m-i*l)/3}*3*i*(2(m-i*l)+i+1)/2)$, which is of order $O(3^m)$, thus requiring very large computational time to decipher the plaintext. Also, the dynamic nature of nodes does not allow brute force attacks to become successful due to large number of possible permutations. Further the brute force attacks fail in this scenario because the pattern that is to be spliced off varies with the plaintext.

The simulations are performed by using C++ Developers compiler on Windows Vista (Home Edition) system. The hardware configuration of the machine used is Core2duo processor/ 2 GB RAM/ 4 MB cache. The results of simulation have been summarized in Table1 and Table 2.

Table 1 shows the performance of proposed algorithm with different sets of plaintext varying in context and length.

Table2 shows the performance of the proposed algorithm on different sets of data, highly diverse in nature covering wide range of Extended ASCII characters.

Table 1: The performance of application with different length of plaintext

Dataset	Size of Plaintext (bytes)	Size of Cipher Text (bytes)	Size of Keys (bytes)	Encryption Time (ms)	Decryption Time (ms)
Data 1	10	12	63	7.8	3.8
Data 2	100	124	555	8.53	6
Data 3	1000	1140	5720	16	12.5
Data 4	10000	11600	57000	72.67	64.13
Data 5	100000	110000	582000	534.13	548.93

Table 2: The performance of algorithm with plaintext of different context

Data Set	Description	Number Of Different Characters	Success Rate of Decryption (%)
Data 1	Only alphabetical characters	52	100
Data 2	Only numerical characters	30	100
Data 3	Only special characters	33	100
Data 4	Combination of characters	80	100
Data 5	Combination of characters	150	100

The observations from the simulation have been plotted in Figure 2 and Figure 3 to carry out the length-time analysis and length analysis respectively.

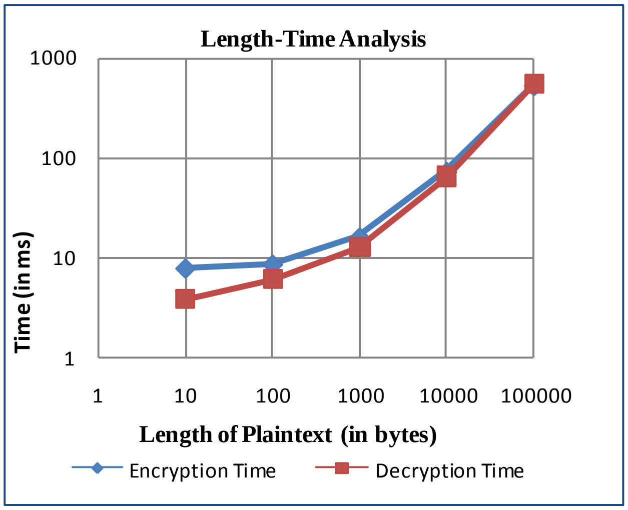

Figure 2: Length Time Analysis

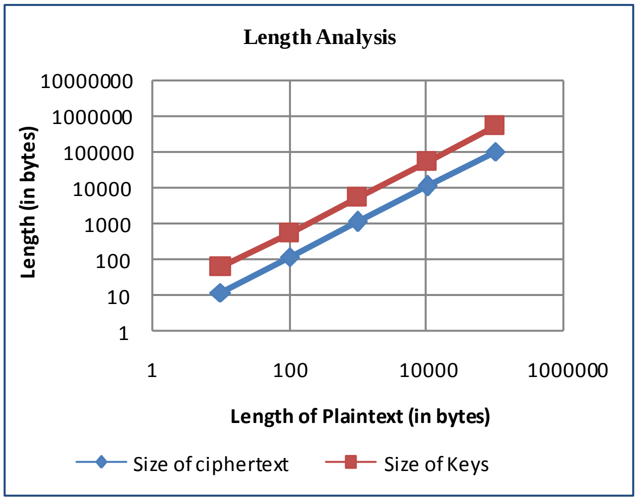

Figure 3: Length Analysis

From the Length-Time analysis as shown in Figure 2, it can be observed that as the length of plaintext increases, the time for encryption and decryption also increases. The encryption time and decryption time are almost same at higher values of the length of plaintext.

From the Length analysis as shown in Figure 3, it is observed that the length of cipher text is 20% more than the plaintext and the length of the keys is 5.5 to 6.5 times the length of plaintext, incorporating sufficient amount of redundancy. However, the length of keys can be reduced by varying the style of splicing.

One of the limitations of mobile networks is that they have limited computational abilities, so the implementation of our algorithm to ensure security can be challenging. On the basis of findings and analysis, the algorithm is found to possess some limitations as follows:

1. During encryption process, there might not be enough introns to be spliced off. A solution is for sender to prepare many starting and ending codes of introns, and select a pair which can result in an appropriate cut off.
2. The complexity of decryption process increases as the size of key increases.
3. Trust authority is required to verify the node entering the Mobile Network.

5. CONCLUSION AND FUTURE WORK

Mobile networks are gaining popularity in large community of people including the research scholars and business enterprises. The vulnerability of mobile networks to attacks makes security one of the major issues in data transmission. The proposed algorithm is analyzed to be strong enough as the permutations required by a brute force attack are sufficiently high to decipher the message being sent across the mobile network. It can be concluded from the various analysis that the proposed DNA-based cryptosystem promises to be a better solution for implementation in securing the mobile networks. Further, this method can be incorporated as a hardware solution. However, the limited computational ability of the nodes in mobile networks is still an issue, which can be worked upon in future.

6. REFERENCES

[1] A. K. Verma, Mayank Dave, R.C. Joshi, “Securing Ad hoc Networks Using DNA Cryptography”, IEEE International Conference on Computers and Devices for Communication (CODEC06), pp. 781-786, Dec. 18-20, 2006.

[2] Ashish Gehani, Thomas LaBean and John Reif. *DNA-Based Cryptography*. DIMACS DNA Based Computers V, American Mathematical Society, 2000.

[3] Behrouz A. Forouzan, “Cryptography and Network Security”, Special Indian Edition, TMH Inc., New York, Chapter 1, pp. 2-13.

[4] Creighton T. Hager, Presentation on “Mobile Ad Hoc Network Security”, Integrated Research and Education in Advanced Networking, 2002 Research Workshop, May 4, 2002 available at http://www.irean.vt.edu/research_workshop_may2002/06_Hager.pdf last accessed on February 26, 2008.

[5] Giancarlo Pellegrino, “Security Analysis of MANET in NS2”, Mini Workshop on Security Framework 2006, Catania, December 12, 2006.

[6] Harvey Lodish, Arnold Berk, Paul Matsudaira, Chris A. Kaiser, Monty Kreiger, Mathew P. Scott, S. Lawerance Zipursky, James Darnell, “Molecular Cell Biology”, 5th edition, W.H. Freeman & Company, Chapter 4, pp. 101-145.

[7] Imrich Chlamtac, Marco Conti, and Jenifer J.-N Liu, “Mobile Ad Hoc Networking: Imperatives and Challenges,” J. Ad Hoc Networks, Vol. 1, No. 1, pp. 13 – 64, 2003.

[8] Samian and Mohd Aizaini Maarof, “Securing MANET routing protocol using trust mechanism”, Normalia Postgraduate Annual Research Seminar 2007, 3-4 July 2007.

[9] www.igd.fhg.de/igd-a8/publications/flyer/manet-security-flyer-english.pdf last accessed on February 26, 2009.