Inconsistency in DNA computing and It's use in Cryptography

Ankita Srivastava
Department of Computer Science and Engineering
NIT, Patna
Patna, India.
1991.aankita@gmail.com

Vivek Pandey
Department of Computer Science and Engineering
NIT, Trichy
Tiruchirappalli, India.
Meet.vivek.pandey@gmail.com

***Abstract*— The computational and storage limitations with silicon computers have propelled computer scientists to search for new dimensions in computer science. DNA computing emerges out to be a very promising field. The use of DNA strands would enable us to do complex calculations in seconds, which would have otherwise needed years. The volumes of data that can be stored have reached a new limit. Recently researchers of Harvard crack DNA storage and have been able to cram 700 terabytes of data into a single gram of DNA strand While developing any system a lot of design issues have to be taken into consideration and same is applicable for DNA computers. The paper makes an effort to deal with the consistency issue of DNA computers. A proof of inconsistency is provided and ground rules to a few DNA based cryptosystems are indexed which take advantage of the inconsistency and make use of it for data security.**

Keywords— DNA computers, Silicon computers, DNA based cryptosystem, PCR.

I. INTRODUCTION

Human body has trillions of data stored in a single strand of Deoxyribonucleic acid (DNA) and the speed with which this data is processed is surprising. Computer scientists are trying to take an inspiration from this sophisticated technology in our body to develop a computer system which may be able to store and process trillions of data with lesser requirements than silicon computers. DNA computers all together come in a different dimension than the silicon computers. They have a totally new method to perform calculations which involves test tubes; enzymes etc. and incorporate biology, chemistry, biotechnology and finally computer science. Rozenberg described a classification in which (i) first type of Computer scientists are who work on theoretical aspect, (ii) second type are more concerned with working on DNA computers and their feasibility [16] in their huge sophisticated biotechnology or biochemical lab.

Deoxyribonucleic acid (DNA) molecules are informational molecules encoding the genetic instructions used in the development and functioning of all known living organisms and many viruses. James D. Watson and Francis Crick established that most DNAs have a double helical structure. DNA is well-suited for biological information storage, since the DNA backbone is resistant to cleavage and the double-stranded structure provides the molecule with a built-in duplicate of the encoded information.

DNA computing [9] is fundamentally similar to parallel computing. It takes advantage of the many different molecules of DNA to try many different possibilities at once. For certain specialized problems, DNA computers are faster and smaller than any other computer built so far. Furthermore, particular mathematical computations have been demonstrated to work on a DNA computer. As an example, Aran Nayebi has provided a general implementation of Strassen's matrix multiplication algorithm on a DNA computer, although there are problems with scaling. In addition, Caltech researchers have created a circuit made from 130 unique DNA strands, which is able to calculate the square root of numbers up to 15.

This paper is organised in three sections one giving a brief overview of how DNA algorithms work to solve a problem. Second section discusses a design issue in DNA computing and third suggests how to use this design concern to develop certain DNA based cryptosystems and thus outlines few basic implementation ideas.

II. ADLEMAN'S CONTRIBUTION

Adleman, a computer scientist at the University of Southern California, came to the conclusion that DNA had computational potential. Adleman is often called the inventor of DNA computers. Adleman [1] in a 1994 issue of the journal Science outlined how to use the DNA to solve directed Hamiltonian Path problem, also known as the "travelling salesman" problem. He gave the DNA solution for "the travelling salesman problem" containing seven nodes.

The Adleman DNA computer created a group of possible answers very quickly, but it took days for Adleman to narrow down the possibilities. Another drawback of his DNA computer is that it requires human assistance. The goal of the DNA computing field is to create a device that can work independent of human involvement.

Adleman [1] paved a way for researchers to explore more about DNA and capability to store more information and processing speed and to utilise the properties to make more powerful computer system than the world has today.

978-1-4673-4529-3/12/$31.00 ©2012 IEEE

A. Solution of Hamiltonian Path problem by Adleman

Problem: A directed graph G, with nodes f, t is provided. The problem is to find whether a Hamiltonian path exists for the graph i.e. is there a directed path visiting every node exactly once from f to t in G.

Solution: Adleman illustrated his DNA algorithm on a graph with 7 nodes but for combinatorial purpose a simple example of a square graph with four nodes is considered. It's assumed here that A is f and D is t.

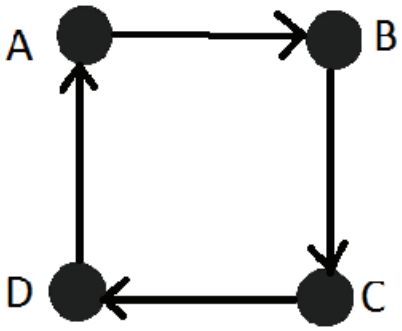

Fig. 1. A graph with four nodes where node A is equivalent to f and node D is equivalent to node t.

1) Adleman's DNA algorithm [1] [6]: The algorithm proposed by Adleman is as follows

Step 0: Represent nodes, edges, paths with DNA.
Step 1: Fill the tubes with all possible paths.
Step 2: Select path from f to t.
Step 3: Select path of correct length.
Step 4: Select path without duplicate vertices.
Step 5: If anything remains
  return "yes"
else
  return "no"

2) Explanation of steps [1] [6]: The explanation of each step is given below

Step 0: Representation - Represent each node with a sequence of 20 random base pairs .It should satisfy these properties
(i) Long enough not to bind each other.
(ii) Short enough for Polymerase chain reaction (PCR) [15] to work
So say node A can be represented as
GCCTAAGCTATTTGGCCAGT
To represent edge, take 10 base suffixes from node U and 10 base prefix from Node V to form an edge S_{UV}
So say if node B is represented as
TGTGCTATGGGAACTCAGCG
So edge S_{AB} will be represented as
TTTGGCCAGTTGTGCTATGG

Step 1: Fill tubes with all paths-Amplify tubes of S_U and S'_U from each node U. Amplify tubes of S_{UV} and S'_{UV} for each edge UV. Mix all tubes into tube T. Overlapping segments will bind each other and leave "sticky ends" to promote further binding.

Steps 2: Select candidate paths f….t.

Step 3, 4: Run PCR in tube T using S_F and S'_F as primers but products in T'. Separate strands with 20n+10 base from T. Put products in tube R. Gel electrophoresis process is used to sort the strands according to the size.

Step 5: If any DNA is left in tube return YES

Else return NO.

III. Inconsistency As Design Issue In DNA Computers

While designing any machine or system one of the most important issues that have to be considered is the consistency of the machine. If the same operation is performed on the machine under similar conditions, is the machine giving the same output? For example a dice is not a good machine considering the consistency because every time a dice is tossed even under similar conditions same output has a less probability.

Designing a DNA computer requires dealing with an important question i.e. are DNA computers consistent i.e. if the same algorithm is run on the same input again and again under similar conditions, will it produce the same result as the output of the algorithm?

To understand this it's important to understand the functioning of each step in the algorithm. Consider the algorithm is run for Hamiltonian path in a DNA computer and all the steps are same as Adleman suggested.

Step 0: Representation of nodes, edges and vertices will be consistent under similar conditions i.e. when same input is used. So this step is consistent.

Step 1 and step 2: These two steps are manual and as long as the inconsistency is not due to the operator these steps are consistent as well.

Step 3 and step 4: These two steps involve the use of PCR reaction. This reaction is controlled by enzymes. It is not under direct control of the operator. Although one can have some control over the rate of reaction, amount of output obtained etc. but there are limits to this control, one cannot control which one of the many possible outcomes should appear as the final output.

It's similar to a die situation although this is more consistent than dice. For small number of faces of a die one can bias the output but for larger number of faces biasing is not an appropriate option. Similarly for small number of possible outcomes it is possible that one can apply some technique and obtain the exact output as intended but for larger number of valid outputs choosing one is difficult. This concept is proved in next section.

DNA are found in cells of living beings and as the inspiration to build a DNA computer is drawn from the activities of DNA in living cells , a similar analogy can be drawn while studying consistency of DNA computers . DNA controls all physical attributes [7] of a person i.e. the eye color, hair color, height,

weight etc. and therefore asking the question whether a DNA computer is consistent is similar to asking the question that if a person aged twenty is allowed to rollback in time just as a software to say 3 time periods

1) Just after the formation of fertilized egg.

2) Just after few cell divisions.

3) A year after birth.

and is allowed to grow back to twenty years again, will all the three cases lead to the same person in all aspects that are controlled by DNA ,which he/she was before the roll back? That is will the person have same eye color, hair color, height and weight as he/she had before the roll back?

The answer is probabilistic. Various factors would act on it. Neglect environmental conditions, even then there might be differences. Same is the case with DNA computers. Suppose a set of DNA strands and a problem say Hamiltonian path problem is chosen and a DNA algorithm is used to determine the Hamiltonian path. The result is obtained and then the output DNA strands are broken up into the constituent DNA single strands which are used as input and further the same algorithm is performed on the same set of DNA strands .The output obtained in all cases might not be same. Every repetition of an algorithm over a problem might return a different output

This probability that every time output might be different presents the design issue the paper is focusing on i.e. "Inconsistency in DNA Computers". Thus a DNA computer would inherit the property of inconsistency. Designers would have to implement certain modifications and specifications while designing a DNA computer.

IV. Proving The Inconsistency In Laboratory

The basic principle used in proving the inconsistency is by making use of the different isotopes [5] of H, C, N and O. Isotopes are variants of a particular chemical element. All isotopes of a given element share the same number of protons; however each isotope differs from the others by its number of neutrons. Therefore isotopes [5] of an element have same chemical properties but different physical properties. This difference in the physical properties is used to differentiate isotopes form one another.

Now the same square Hamiltonian graph is taken under consideration upon which the Hamiltonian algorithm was applied. This time there are two sets of input DNA strands. One of these two sets contains all pure normal molecules. The word normal means there are no isotopes of any element present in any of the molecules constituting the input DNA strands. The four nodes of the graph formed by normal molecules are labelled as A, B, C, and D. The second set contains molecules which have pure isotopes of Carbon, hydrogen, oxygen, nitrogen etc. Only sticky ends here have isotopes in the molecules forming the input DNA strand as anyways the molecules which occupy the centre position do not take part in PCR and hence their constituents are not of much importance in this experiment. These nodes i.e. one containing pure isotopes are labelled as A^*, B^*, C^* and D^*. The two sets of DNA strands are mixed in equal proportion i.e. both pure normal and pure isotopic DNA strands are each present in the mixture in a ratio of 1:1.

The problem again is to determine a Hamiltonian path .Under normal circumstances in the absence of isotopes the DNA algorithm would give a solution [1] [6] [3] as ABCD where ABCD represents a directed path from A to B to C to D i.e. a directed path from A to D via B and C. But now since isotopes are also present in 50% of the DNA strands other solutions corresponding to this problem will be obtained as well .These different solutions though would be chemically same but will have different physical properties [8] and can be differentiated. The number of solutions that would be obtained depends on number of nodes in the graph. Each node can have either pure normal DNA strand or pure isotopic DNA strand and thus each node represents 2 possibilities. Thus if there are N nodes 2^N outcomes are possible. In present case a graph of N=4 is being considered and hence 16 outcomes are possible in all.

These solutions are

ABCD (directed path from A to B to C to D i.e. a directed path from A to D via B and C)

$ABCD^*$

ABC^*D

ABC^*D^*

AB^*CD

AB^*CD^*

AB^*C^*D

$AB^*C^*D^*$

A^*BCD

A^*BCD^*

A^*BC^*D

$A^*BC^*D^*$

A^*B^*CD

$A^*B^*CD^*$

$A^*B^*C^*D$

$A^*B^*C^*D^*$

Thus there are 16 possible solutions corresponding to the same problem. It is to be noted that all the combinations or outcomes that do not give a valid Hamiltonian path have been removed manually and only the ones which give a correct solution to the Hamiltonian problem [1] [6] [3] are being considered. Paths like ABBD etc. are not valid Hamiltonian

paths and thus have been removed. The 16 outcomes that have been obtained are all valid. Also it is to be noted that ABCD, BCDA, CDAB, DABC are all variants of same solution and therefore they are not being considered here for sake of clarity. Ultimately these 16 outcomes are possible for a small graph considered here. If a large graph of say having 10000 nodes is considered then the number of possible outcomes would be huge even though if the same variants are counted as 1. In the cases where nodes are connected in such a way that variants are not the same the number increases to a much larger extent.

The probability of getting exactly ABCD as the solution for the graph with four nodes has decreased from 1 to 1/16. Similarly for larger graphs this will decrease even to a larger extent. Therefore differentiating these different solutions so as to get the same solution every time would be practically a very tedious job.

In next step of the experiment pure paths i.e. ABCD and $A^*B^*C^*D^*$ are removed from the test tube containing solution and the DNA strands are broken back to the constituent single DNA strands .The same experiment i.e. applying DNA algorithm to find the Hamiltonian path is repeated again upon these strands. Earlier there were 16 solutions corresponding to 1 problem. Now there are only 14 as 2 have been removed. Here by the context that there will be 14 solutions it is the ratio that is intended and not the actual number. Which 14 out of the 16 combinations will be formed depends on many factors and hence is probabilistic.

The probabilistic nature that it cannot be inferred for sure which solution of all the possible solution will be obtained in the case when only one solution is desired gives DNA computer the property of "Inconsistency". Since the reaction will be controlled by enzymes it would be hard to find which solution has more probability of appearing in the test tube. For large scale problems the inconsistency would be pretty high.

Thus this design issue that is inconsistency has to be taken in consideration while designing DNA computers. Here it is not intended to mean that there is no possible way to make DNA computers consistent. This is a similar case where there are a lot of design issues while designing a silicon computer but that have been overcome by various modifications and involving new technologies.

Also this paper discusses very important use of inconsistency in DNA computers i.e. use in providing a secure communication channel and guidelines to certain very strong and potential cryptographic algorithms is provided.

V. Using Inconsistency In Cryptography

The design issue that the paper points at i.e. the inherent inconsistency in the DNA computers can be put to a lot of applications [2] [3] [4] .One such application is its use in a DNA based cryptosystem. As the paper states that if same algorithm is repeated on same set of inputs under similar conditions the results obtained may not be same, and as the size of input set increases time required for computation grows exponentially, it becomes a hard problem to apply brute force and determine which one of the several solutions is the solution we are concerned with. This property thus can be used as the basis of a cryptographic algorithm.

A. Cryptosystem using one time hash functions.

The idea of inconsistency can be used to design a cryptosystem involving one time hash function (pad). As the output is probabilistic and every time a new output can be obtained, this property is similar to using one pad. A one-time pad [12] corresponding to one input would be generated every time by applying some algorithm say Directed Hamiltonian path problem on a DNA computer and then used, and when that pad dies or is invalid a new pad for subsequent communication would be required.

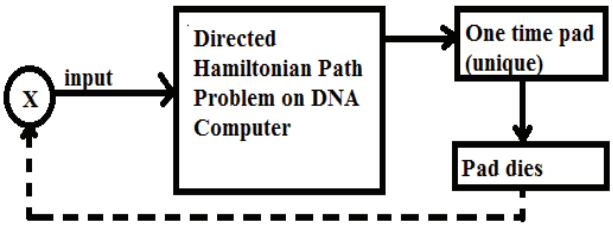

Figure 2. A symbolic representation of the one-time pad generation cryptosystem.

Since next time another set of pad is required so the DNA algorithm will be applied again to get an output, and as the paper states the chances of having a new output every time and thus a newpad is very high.

B. Cryptosystem using a third party

A second cryptographic design can be communication involving third party X and the concept of certificates i.e. establishing a secure communication channel between the parties wanting to communicate and then using this secure channel for further communication. In this scheme a reliable third party is present whose task is to manage certificates. Now it's assumed there are N numbers of companies who have endorsed to use the service provided by the reliable party. Here company may refer to a large or small company or even an individual. This system is transparent and thus it doesn't matter whether the company here is single entity or some organized group. Now each company would choose one NP hard problem say Hamiltonian path or Subset sum problem. Also each company will generate a set of correct solutions for the problem and choose a subset of the solutions each solution acting as a certificate. The certificates are transported to the third party using various certification transmission methods already existing.

Now say a party A wants to communicate to another party B. A will contact the third party and asks for B's certificate. Third party would check if A is the authenticated member, if yes then it would take the set which has all the certificates of B, randomly permutate it and send the certificates in an

encrypted form to A. An existing DNA algorithm can be used to encrypt the certificates. Also the third party would send the permutated sequence to B stating that the party A queried about its certificates. B knows thus that A has requested for its certificates meaning A might need to communicate with it. Therefore it would wait for a message from A.

A starts sending a sequence of message each with B's certificate attached to it in the order the third party had provided it. B would check if the certificates are in same order, if yes then would send an acknowledgement to A and the process will continue until entire set has been transmitted and a secure channel is established.

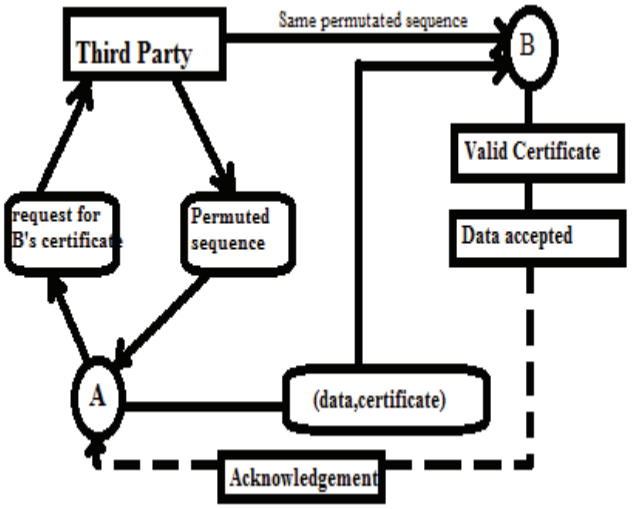

Fig. 3. A symbolic representation of cryptosystem using third party.

Else if even one of the certicates transmitted is not in correct order B would report an error and the communication channel would be terminated. Also as soon as B receives a notification from third party that A queried for its certificates, B would start a timer, if a request from A for establishing a communication channel doesn't come within a fixed time interval the timer times out and any further communication of A with B would require asking for a new permutation of B's certificates from third party.

C. Cryptosystem using public key

Yet another scheme may be to use inconsistency in encryption-decryption algorithms. Suppose A wants to communicate to B same as before but no third party is present now and a secure channel needs not to be established before communication. So encryption and decryption scheme is used. Same as before A and B would choose a NP hard problem Say A chooses Directed Hamiltonian problem and B chooses Subset sum. The size of input and the NP hard problem that is being used would be made public, i.e. if A chooses Directed Hamiltonian path problem and input size of 10000 nodes then it would announce N=10000 and Directed Hamiltonian problem as its public key . Similarly if B uses Subset sum problem and input Size as 100000 then B would announce N=100000 and Subset sum as its public key.

A and B will solve their NP hard problem and generate a set of solutions. Each of them would select a particular subset of the solution and treat it as set of private keys. A set of private keys and not just one private key is to ensure a more reliable and safe cryptosystem.

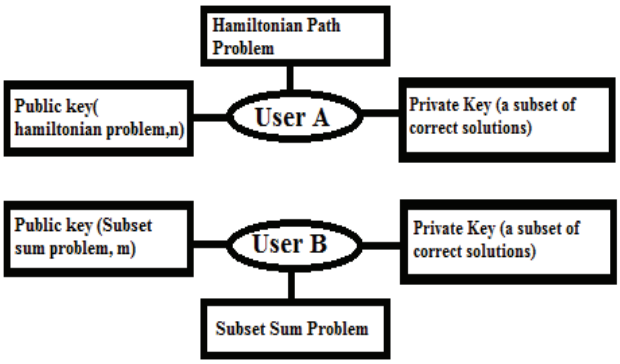

Fig. 4. A symbolic representation of cryptosystem using public and private key.

Following this scheme of public and private keys a cryptosystem to encrypt a message using the public key of the receiver and to send it can be generated. On receiving the receiver would apply a specified decryption algorithm and use his private key to get back the plaintext.

Thus the inconsistency can be put to use for developing a strong cryptosystem or to establish a highly secure communication channel or to use in one time padding

VI. Conclusion

The DNA computers are potential future of computing, and it's one of the most intriguing areas for the scientists, be it in field of computer science, biotechnology or biochemistry. The paper discusses about the DNA algorithms work to solve a problem and then further talks about the design issue in DNA computing. However by discussing the design issue i.e. inconsistency occurring during computation, the paper only aims to seek attention to this aspect of DNA computing and by no means spread negativity about this very potential field of computing.

Therefore further the paper gives an outline to some very efficient DNA based cryptosystem, which make use of the inconsistency factor of the DNA algorithms. The inconsistency makes DNA algorithms a NP-hard problem and thus they become very apt for cryptographic algorithms

As a conclusion, DNA computing is a very promising field and can overcome many limitations of silicon computers. Only by delving further into the field, DNA computers can be brought to an existence. The use of inconsistency, which is an

integral part of DNA algorithms today, we can better use DNA computing. However there is hope that sooner, more efficient experimental scenarios will come into implementation which may reduce these inconsistencies when not desired.

REFERENCES

[1] L M Aldeman "Molecular Computation of solutions to combinatorial problems " Science Volme 266 No. 5187 1994

[2] Junzo Watada , Rohani Binti abu Bakar " DNA computing and its applications " IEEE Eighth International Conference on Intelligent Design and applications

[3] JY Lee , SY Shin, TH Park and B-T Zhang "Solving travelling salesman problems with DNA molecules encoding numerical value "Biosystems Vol 78

[4] A Gehani ,T La Bean and J Reif " DNA based cryptography " Lecture Notes in Computer Science

[5] http://www.colorado.edu/physics/2000/isotopes/index.html

[6] James A Foster "Computing with DNA " 1997K Tanaka , A Okamota and I Saito "Public Key Systems Using DNA as one way function "Biosystems vol 81

[7] Olgica Milenkovic , Navin Kashyap , Bane Vasic "On DNA computers controlling gene expression levels" IEEE 44th conference

[8] P Decelles "Control of gene expression "

[9] Junzo Watada, Rohani binti abu Bakar "DNA Computing and Its Applications" Eighth International Conference on Intelligent Systems Design and Applications.

[10] Chunde YANG Guohui WEI, Jun TAN Jingbo XIE "The Breadth First Search Traversing Algorithm of the Graphs in DNA Computer".

[11] Miki Hirabayashi, Akio Nishikawa "Analysis on Secure and Effective Applications of a DNA-Based Cryptosystem" 2011 Sixth International Conference on Bio-Inspired Computing: Theories and Applications.

[12] Ralph C. Merkle "One Way Hash Functions and DES"

[13] www.wikipedia.org

[14] www.howstuffworks.com

[15] D. Rooss, "Recent Developments in DNA-Computing," Proceedings of the International Symposium on Multiple-Valued Logic, 1997, pp. 3-9.

[16] G. Rozenberg and A. Salomaa, "DNA computing: New ideas and paradigms," Lecture Notes in Computer Science (LNCS), Springer-Verlag, Vol. 7, 2006, pp. 188- 200.