# A Novel Multi-Client Authentication Method Using Infection of Bacteria

Rezvan Dastanian, Arash Karimi, and Hadi Shahriar Shahhoseini

***Abstract*—The massive parallelism and data hiding capabilities inherent in DNA strands make them innovative media to base security primitives. In this paper we propose an authentication scheme based on infection of Escherichia coli bacteria. Our scheme is the first illustration of using DNA computing techniques in authentication. Our proposed technique simply simulates a client-server network and each one of the clients keeps a test tube containing their encoded identity. We have conducted a security analysis for our scheme, which demonstrates that it is hard for an adversary to extract users' identities. Our scheme can be easily implemented using genetic engineering techniques and with the growth of utilizing genetic engineering techniques in computing, it will supersede the traditional trends for authentication of clients in networks.**

***Index Terms*—Authentication, E. coli bacteria, bacteriophag, DNA computation.**

## I. Introduction

Adleman in his groundbreaking paper [1] revealed the potential of DNA molecules to solve computationally hard problems and thus founded the interdisciplinary area of DNA computing. Since then, a large number of papers tried to extend his tiny DNA computer [2]-[5] and other magic capabilities of DNA molecules such as their ability to hide information were demonstrated in subsequent papers (e.g. [6]). The demanding task of providing security in large scale networks combined with the massive parallelism of DNA molecules and their capability to store vast amount of information swayed the research direction in DNA computation to provision of security primitives using DNA molecules. In this respect, implementation of the only information-theoretically secure cipher, the Vernam One-time pad scheme, using DNA molecules was proposed by Gehani et al. [6]. A molecular computer scheme to break The Data Encryption Standard [7] based on in-vitro synthetic DNA manipulation was proposed by Boneh et al. and afterwards by Adleman in [8] and [9] respectively. DNA chip-based implementation of a steganography scheme was proposed in [6]. Shapiro et al. proposed the first practical implementation of an in-vivo finite automaton in [4] which is able to distinguish between strings having odd number versus even number of input symbols. This is the first step towards implementing a Turing machine using biological cells. We previously proposed a Watermarking scheme based on molecular manipulation of E. coli bacteria which can be implemented on an in-vivo computer based on Escherichia coli bacteria [5].

In this paper we present a multi-client authentication method in which each client can be authenticated using his/her personal identity. Our proposed sample network is a client-server network and has a ring topology. We provide mathematical formulations for cut and recombinant systems and present a security analysis to support our ideas.

The rest of the paper is organized as follows. In section II, we present some preliminary definitions. In section III, the proposed scheme is presented. In section IV, we have evaluated our proposed scheme. In section V, we have conducted a security analysis. In section VI conclusions are drawn.

Manuscript received April 15, 2012; revised May 13, 2012. This work was supported in part by Iran Telecommunication Research Center (ITRC).

The authors are with the Department of Electrical Engineering Of Iran University of Science and Technology, Narmak, Tehran, 16846-1311, Iran (e-mail: r_dastanian@elec.iust.ac.ir, ar_karimi@elec.iust.ac.ir, hshsh@iust.ac.ir).

## II. Preliminary Definitions

Since Recombinant behavior of DNA is mathematically modeled with strings of formal language theory as the basic data structure, and in order to justify our proposed scheme, we need to recall some basic notions from the theory of formal languages first. (for more information, the reader is referred to [10]).

### A. Recombinant Systems

We denote the alphabet of a language by a finite set $V$. In this regard, the free monoid generated by $V$ under the operation of concatenation is denoted by $V^*$ and the empty string is denoted by $\lambda$. The set $V^* \setminus \{\lambda\}$ is denoted by $V^+$. A multiset over A is a function $F$ represented in (1):

$$F : A \to N \bigcup \{\infty\} \tag{1}$$

$F$ Illustrates the number of copies of $x \in A$ present in the multiset $F$.

A molecular system can be mathematically modeled as a quadruple $\sigma = (O, O_T, P, A)$, where $O$ and $O_T$ are sets of objects and terminal objects respectively. $P$ is a finite set of productions and $A$ is a finite multiset of axioms from $O$. The language generated by $\sigma$ is shown in (2):

$$L(\sigma) = \{w \in O_T \mid A \Rightarrow_{\sigma}^{*} L, (L, w) \geq 1\}. \tag{2}$$

The result of a computation over $\sigma$ is a sequence of $L_i$ such that $0 \leq i \leq n$, $n \geq 0$.

For two arbitrary multisets $L$ and $L'$ we say that a production $p \subseteq P$ produces $L'$ from $L$, if and only if

$L' = (L - (u_1 + ... + u_k)) + (v_1 + ... + v_m)$ holds true for some $u_i, v_i \in 0$ with $(u^k, v^k) \in p$ and at the same time we should have $L_i \Rightarrow_\sigma L_{i+1}$ for $0 \le i \le n$.

*B. Cut and Insertion Systems*

A cutting system [11] can be formally described as a 3-tuple $\theta = (\Delta, M, C)$ where $\Delta$ is an alphabet, $M$ is a finite set of markers, $C$ is set of cutting rules of the form $c = u\#m\$n\#v$, where $u \in \Delta^* \bigcup M\Delta^*$, $v \in \Delta^* \bigcup \Delta^* M$ and $n, m \in M$. We should note that $\#$ and $\$$ are special symbols not in $\Delta \bigcup M$ for $x, y, z \in \Delta^+ \bigcup M\Delta^* \bigcup \Delta^* M \bigcup M\Delta^* M$ and the cutting rule $c$ defined as above, we define $x = \alpha u v \beta$ and $y = \alpha u l$ and $z = m v \beta$. We define concatenation of the words $x$ and $y$ as combination of them. So, we define $z$ as the concatenation of $x$ and $y$, as demonstrated below: $x' \overset{\Delta}{=} conc(y, z) = aulmv\beta$. If we compare strings $x$ and $x'$, we can observe that we have inserted the substring '$lm$' into the string $x$ to derive $x'$.

As we have seen above, we can model the process of cutting and insertion of a DNA sequence into another DNA sequence by means of our language theoretic notions. Assume that we have DNA sequence of E. coli bacteria shown as $x = x_1 u_1 u_2 x_2$. By adding a restriction enzyme to $x$ we can break the substring $u_1 u_2$ and then according to the procedure explained above, we can insert a sequence $lm$ between $u_1$ and $u_2$ and derive $x' = x_1 u_1 l m u_2 x_2$.

The above process takes place in the infection of E. coli using bacteriophages. Such that in $x'$, the substring $lm$ is the genome of the bacteriophage and $x$ is the genome of E. coli.

## III. The Proposed Scheme

Our proposed model to authenticate a client communicating with the server consists of infection of E. coli bacteria with a variety of bacteriophages such that each client in order to be authenticated for the server, keeps a test tube containing a specific enzyme to extract their bacteriophage from E. coli bacteria genome, while the test tube containing E. coli bacteria is kept with the server. The overall process to encode names or encrypted version of identity of a client is depicted concisely in "Fig.1" and it can be described as follows:

First, determination of a specific gene from genome of a bacteriophage in which the length of DNA is at least three times as long as the length of ASCII code or encrypted version of identities of clients.

Second, the ASCII code or encrypted version of the clients should be determined.

Finally, the encoded bacteriophage should be kept in a test tube.

The illustrated steps shown in the flowchart of "Fig.1" should be repeated for all clients that are authorized to connect to the server using different bacteriophages or plasmids such that if there are $n$ clients, there will be $n$ test tubes containing encoded bacteriophage or plasmid. The contents of all test tubes will be then added to the test tube containing Escherichia bacteria such that eventually there will be a single contaminated bacteria.

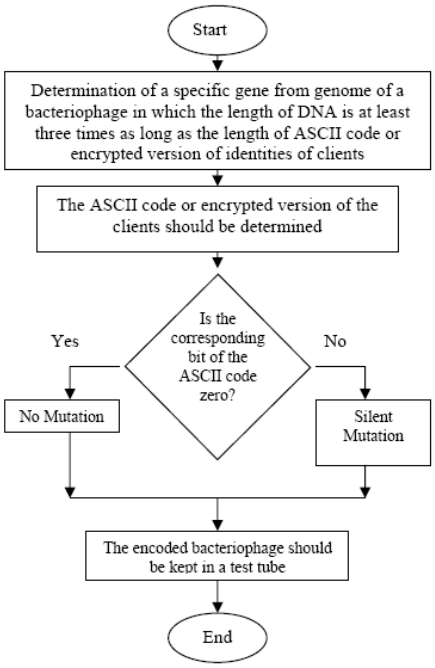


Fig. 1. The procedure of encryption-hiding the identity information in a bacteriophage.

In order to encode the identities of clients in the bacteriophage, we will use the multiform property of codons in the genetic code table. Indeed, a specific amino acid can be resulted from more than one codon. In fact, each codon consists of three nucleotides each of which is selected from the alphabet $\{A, C, G, T\}$ and there will be 64 possibilities for codons, while the overall of 20 amino acids exist. So, each amino acid may have been produced by more than one codon. This will cause that some kind of mutation (the so-called silent mutation) in codons will not change the corresponding translated amino acid of that codon. Thus no change will be made in the protein translated by that codon. Due to some exceptions in genetic code table as shown in table I, we should note that the considered region of the selected gene should only contain the multiform codons.

To proceed with the authentication process, each client adds the contents of his/her test tube, which contains the enzyme that corresponds to his/her identity encoded into genome of a plasmid or a bacteriophage, to the test tube that contains the infected bacteria. In this way, the infected bacteria is cut from the specific cos site which corresponds with that bacteriophage or plasmid and then the process of centrifuge should take place to extract the identities of clients and the server then compares the encoded bacteriophage or plasmid with the stored bacteriophage and for example in the ASCII code case, the extracted string is read with the block size of eight bits and the character that corresponds to each

block is then determined. Indeed, each client can do this job, since he knows that his identity is encoded into which gene from which bacteriophage in which cos site of that gene.

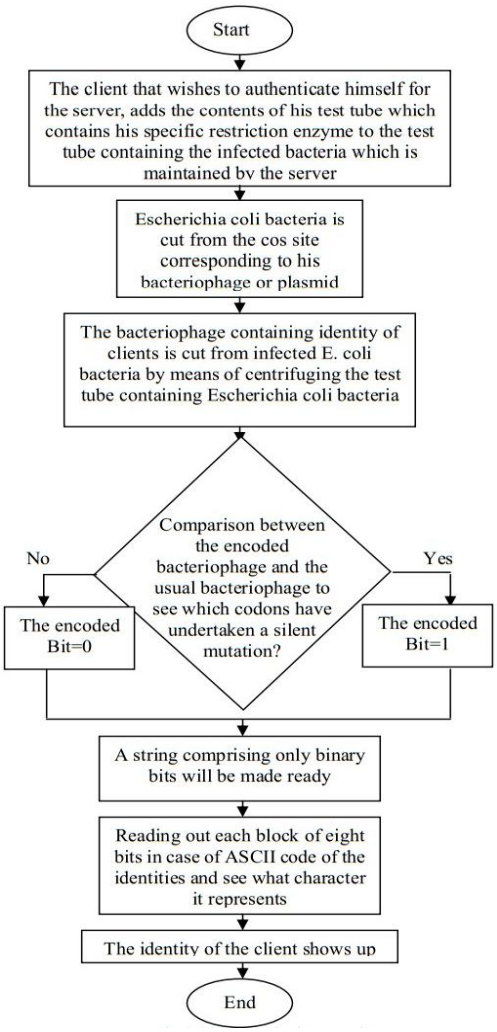


Fig. 2. The authentication procedure

The overall process of authentication is depicted in "Fig.2"

TABLE I: TABLE OF THE GENETIC CODE

| First Position | Second position of Codon: T | C | A | G | Third Position |
|---|---|---|---|---|---|
| T | TTT Phe[F] | TCT Ser[S] | TAT Tyr[Y] | TGT CyS[C] | T |
| | TTC Phe[F] | TCC Ser[S] | TAC Tyr[Y] | TGC Cys[C] | C |
| | TTA Leu[L] | TCA Ser[S] | TAA TeR[end] | TGA Ter [end] | A |
| | TTG Leu[L] | TCG Ser[S] | TAG Ter[end] | TGG Trp[W] | G |
| C | CTT Leu [L] | CCT Pro [P] | CAT His [H] | CGT Arg [R] | T |
| | CTC Leu [L] | CCC Pro [P] | CAC His [H] | CGC Arg [R] | C |
| | CTA Leu[L] | CCA Pro [P] | CAAGln[Q] | CGA Arg[R] | A |
| | CTG Leu[L] | CCG Pro[P] | CAG Gln[Q] | CGG Arg[R] | G |
| A | ATT Ile[L] | ACT Thr [T] | AAT Asn[N] | AGT Ser[S] | T |
| | ATC Ile[L] | ACC Thr [T] | AAC Asn[N] | AGC Ser [S] | C |
| | ATA Ile[L] | ACA Thr[T] | AAA Lys[K] | AGA Arg [R] | A |
| | ATG Met[M] | ACG Thr[T] | AAG Lys[K] | AGG Arg[R] | G |
| G | GTT Val[V] | GCT Ala[A] | GAT Asp[D] | GGT Gly [G] | T |
| | GTC Val[V] | GCC Ala[A] | GAC Asp[D] | GGC Gly [G] | C |
| | GTA Val[V] | GCA Ala[A] | GAA Glu[E] | GGA Gly [G] | A |
| | GTC Val[V] | GCG Ala[A] | GAC Glu[E] | GGG Gly[G] | G |

To proceed with the authentication process, each client adds the contents of his/her test tube, which contains the enzyme that corresponds to his/her identity encoded into genome of a plasmid or a bacteriophage, to the test tube that contains the infected bacteria. In this way, the infected bacteria is cut from the specific cos site which corresponds with that bacteriophage or plasmid and then the process of centrifuge should take place to extract the identities of clients and the server then compares the encoded bacteriophage or plasmid with the stored bacteriophage and for example in the ASCII code case, the extracted string is read with the block size of eight bits and the character that corresponds to each block is then determined. Indeed, each client can do this job, since he knows that his identity is encoded into which gene from which bacteriophage in which cos site of that gene.

The overall process of authentication is depicted in "Fig.2"

## IV. EVALUATION

To evaluate our proposed scheme, we assume that four clients called Alice, Bob, Carol and Debbie want to connect to the server (as shown in "Fig.3") and A, B, D and C in (3) to (6) stand for Alice, Bob, Debbie and Carol respectively. And we have also used different bacteriophages 'FD', '186', 'Lambda' and 'T7' to encode their names respectively.

To proceed with the authentication procedure, the ASCII code of identities (e.g. names) should be determined. (3) to (6) show ASCII code of the clients.

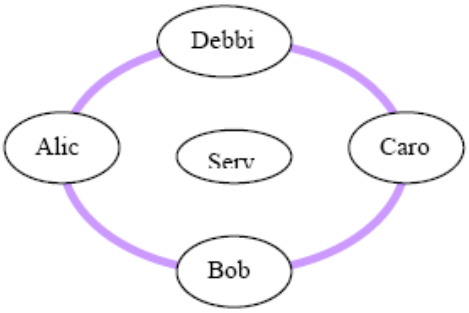


Fig. 3. Our sample client-server network.

A=01000001011011000110100101100011011001001 (3)

B=010000100110111101100010 (4)

D=01000100011001010110001001100010011010001 (5)
01100101

C=01000011011000010111001001101111011011000 (6)

Four different bacteriophages or plasmids are used to encode names of the clients. Each client encodes a specific part from genome of his bacteriophage or plasmid. The relations shown in (7) to (10) demonstrate parts of the selected genes.

fd=
AATGTATCTAATGGTCAAACTAAATCTACTCGTTCGCAGA
ATTGGGAATCAACTGTTACATGGAATGAAACTTCCAGACA
CC
GTACTTTAGTTGCATATTTAAAACATGTTGAACTACAG (7)

186=AATGTATCTAATGGTCAAACTAAATCTACTCGTTCGC
AG
AATTGGGAATCAACTGTTACATGGAATGAAACT (8)

lambda=
TTAAAATACCCTCTGAAAAGAAAGGAAACGACAGGT
GCTGAAAGCGAGGCTTTTTGGCCTCTGTCGTTTCCTTTCTC
T
GTTTTTGTCCGTGGAATGAACAATGGAAGTCAACAAAAAG
CA
GCTGGCTGACATTTTCGGTGCGAG (9)

T7= TTCCCTAAGGGTTGGGGATGACCCTTGGGTTTGTCTT
TGGGTGTTACCTTGAGTGTCTCTCTGTGTCCCTATCTGTTA
CAGTCTCCTAAAGTATCCTCCTAAAGTCACCTCCTAACGT
CC (10)

In ASCII codes of the clients, when we see a bit "1", its corresponding codon undertakes a silent mutation and if we encounter a bit "0", no mutation takes place. The DNA strands in (11) to (14) illustrate this concept in a better way.

cfd= AATGTGTCTAATGGTCAAACTAAGTCTACGAGG
TCGCAAAACTGGGAATCAACGGTGACAAGCAATGA
AACGTCCAGGCATCGTACTTTAGTGGCGTATCTGAA
GCATGTTGAGCTACAA (11)

c186=AATGTGTCTAATGGTCAAACGAAATCTACGAGGT
CGCA
AAACAGCGAGTCAACGGTGACATGGAATGAGACT (12)

clambda=
TTAAAGTACCCTCTGAAGAGAAAGGAAACTACGGG
TGCTGAGAGCGAAGCTTTCAGCCCTCTGTCGTTCCCTTTC
AG
CGTGTTTGTCCGTGGGATGAACAACGGGAGTCAGCAAAA
AG
CGGCTGGGTAACATTTTAGGTGCGAA (13)

cT7= TTCCCGAAGGGTTGGGGATAACCGTTGGGGCTGTCT
TTGGGTGTTACGTTGAGGGTGAGCCTGTGTCCGTATCTGC
TGCAATCTCCGAAGGTGAGCTCCTGAAGGCACCTGCTGA
C
GTCC (14)

All the bacteriophages are added to the bacteria and each bacteriophage enters from its own cos site and DNA of genome of the bacteria remains circular and as the number of clients increases, the length of the circular DNA genome of E. coli increases. The test tube containing the infected E. coli is kept with the server.

Assume that Alice has decided to authenticate herself for the server. So, she adds the specific enzyme in order to extract her phage to the test tube that is with the server.

## V. Security Analysis

In this part we aim to analyze the security of our proposed scheme. Our analysis reveals security of our scheme theoretically which proves that our proposed scheme can produce a circular string that possesses substrings that are hard to be extracted by those who do not have the required information. And meanwhile, we present some factors that enforce the security of our proposed scheme.

Some factors that contribute to security of our proposed scheme are as follows.

1) the bacteriophage that bears the encoded information of identities.

2) the specific gene from genome of the bacteriophage that contains the data.

3) the specific location of the gene that the data is encoded into.

4) the specific sticky end sequence and the length of it that is the cut point of E. coli bacteria genome.

All these four factors play some role in security of our proposed scheme. Based on these parameters, we assume circumstances under which the attack can take place against our system.

Assume that an adversary (say Eve) possesses the test tube containing the infected bacteria. In order to find the encoded identity of her intended victim, present in E. coli, she should analyze the DNA of the genome of E. coli by sequencing the gained solution containing E. coli bacteria. But without knowledge of the location of Alice identity, Eve should search the whole genome of E. coli plus genome of all the bacteriophages that their genome are inserted into genome of E. coli. So in order to mount a brute force attack on our proposed system, the adversary needs to check the whole DNA sequence in the test tube on the server side the length of which makes the brute force attack infeasible.

According to what stated in section II, assuming that there are 'n' clients who want to connect to the server, each of which inserts a sequence $l_i m_i$ as their identity encoded in DNA sequence of some gene from genome of their phage, in the test tube containing the infected E. coli bacteria the sequence $x' = x_1 u_1 l_1 m_1 b_1 l_2 m_2 b_2 .. l_n m_n b_n u_2 x_2$ will be produced in which the $' l_i m_i '$, $1 \le i \le n$ subsequences are not necessarily in the right order shown above and they are permuted and then they are placed in $x'$ and $b_i$ (s) demonstrate parts of the E. coli bacteria genome between bacteriophages $i$ and $j$ next to each other. We can observe that theoretically deriving $l_i m_i$ from this sequence is very hard because this subsequence is indistinguishable from its neighbor subsequences.

As opposed to the adversary (Eve), the clients themselves can easily extract and prove their identities to server because they have knowledge of their bacteriophage and the cos site of standing their information-encoding bacteriophage. In this way she/he uses the required restriction enzyme which has been used to make a stepwise cut in DNA of E. coli bacteria to extract and prove her/his identity to the server. Since there are a variety of restriction enzymes with an extensive spectrum of lengths available in biotechnology and each of which can be utilized to make a cut in a specific site of E. coli bacteria, an adversary finds it difficult to check all parts of the long DNA molecule with all restriction enzymes to derive an encoded (or even maybe encrypted) information that contains identity of her victim.

## VI. Conclusions

In this paper a new authentication method based on infection of Escherichia coli bacteria is presented which can be easily utilized by in-vivo DNA computers. Each client, in order to authenticate himself for the Server, encodes the ASCII code of his identity in a bacteriophage genome using

the silent mutation property of codons that may not cause phenotypic changes and then he will add the ingredients of the test tube containing his encoded identity in the infected E. coli solution which is kept with the Server. And each time any client wants to authenticate himself for the server, he pours the required enzyme in Escherichia coli bacteria solution and then, using the centrifuge technique can extract the bacteriophage containing his encoded identity. After extracting the encoded bacteriophage, any client decodes the gained identity to derive his identity. The security of our proposed scheme is high because each client has knowledge of his own cos site and his own bacteriophage and also the location of the genome that his identity has been encoded. And as shown in section V, it is hard for an adversary to extract identities of the clients. The simulation results shown in section IV depict the correctness of our claims.

## VII. Acknowledgment

The authors are thankful of the financial support provided by Iran Telecommunication Research Center (ITRC).

## References

[1] L. M. Adleman, "Molecular computation of solutions to combinatorial problems," *Science*, vol. 266, 1994, pp. 1021-1024.

[2] L. M. Adleman, "On constructing a molecular computer," In R.J. Lipton, E.M. Baum (Eds), DNA based Computers I, *Proceedings of a DIMACS Workshop*, Princeton, American Mathematical Society, Providence, RI, 1996, pp.1–22.

[3] J. Khodor and D. Gifford, "Design and implementation of computational systems based on programmed mutagenesis," In *Preliminary Proceedings of 4th DIMACS Workshop on DNA Based Computers*, 1998, pp. 101–108.

[4] Y. Benenson, B. Gil, U. B. Dor, R. Adar, and E. Shapiro "An autonomous molecular computer for logical control of gene expression," *Nature*, vol. 414, pp. 430–434, 2004.

[5] A. Karimi, R. Dastanian, and H. S. Shahhosseini, "A New Watermarking Scheme For An in-vivo Computer Based on Infection of E. coli," in *Proceedings of ICCEE*, vol. 8, pp.484-489, 2010.

[6] A. Gehani, T. L. Bean, and J. Reif, "DNA-based Cryptography, Aspects of Molecular Computing," *Springer-Verlag Lecture Notes In Computer Science*, vol. 2950, 2004.

[7] National Bureau of Standards: "Data Encryption Standard," *U.S. Department of Commerce*, FIPS, vol. 46, Jan. 1977.

[8] D. Boneh, C. Dunworth, and R. Lipton, "Breaking DEES Using a Molecular Computer," *Princeton CS Tech-Report CS*-TR-489-95.

[9] L. M. Adleman, P. W. K. Rothemund, S. Roweis, and E. Winfree, "On applying molecular computation to the Data Encryption Standard," *2nd annual workshop on DNA Computing, Princeton University*, Eds. L. Landweber and E. Baum, DIMACS: series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society. pp. 31-44, 1999.

[10] J. Dassow and G. Paun, "Regulated Rewriting In Formal Language Theory," *Springer, Berlin*, 1989.

[11] R. Freund, "Generalized P-Systems with Splicing and Cutting/Recombination," Grammars, vol. 2, no. 3, pp. 189-199, 1999.

**Rezvan Dastanian** was born in Iran, Ahvaz, in 1987. She received the B.Sc. and M.Sc. degrees in electrical engineering from Iran University of Science and Technology, Iran, Tehran in 2008 and 2011 respectively and is currently working toward the Ph.D. degree in electrical engineering at Shiraz University of Technology. Her research interests include cryptography, Biochemical computing, current mode and data converter.

**Arash Karimi** Received the B.S. and M.S. degrees in the department of electrical engineering from Amirkabir University of Technology (Polytechnic of Tehran) and Iran University of Science and Technology (IUST), Tehran, Iran, in 2008 and 2011, respectively. His research interests include cryptography, unconventional methods in computation with a focus on cryptanalysis, Biochemical computing, and formal languages and automata.

**Hadi Shahriar Shahhoseini** received B.S. degree in electrical engineering from University of Tehran, in 1990, M.S. degree in electrical engineering from Azad University of Tehran in 1994, and Ph.D. degree in electrical engineering from Iran University of Science and Technology, in 1999. He is an assistant professor of the electrical engineering department in Iran University of Science and Technology. His areas of research include networking, supercomputing and reconfigurable computing. More than 130 papers have been published from his research works in scientific journals and conference proceedings. He is an executive committee member of IEEE TCSC and serves IEEE TCSC as regional coordinator in middle-East Countries.