DNA-based Cryptography

Ashish Gehani, Thomas H. LaBean, and John H. Reif

ABSTRACT. Recent research has considered DNA as a medium for ultra-scale computation and for ultra-compact information storage. One potential key application is DNA-based, molecular cryptography systems. We present some procedures for DNA-based cryptography based on one-time-pads that are in principle unbreakable. Practical applications of cryptographic systems based on one-time-pads are limited in conventional electronic media, by the size of the one-time-pad; however DNA provides a much more compact storage media, and an extremely small amount of DNA suffices even for huge one-time-pads. We detail procedures for two DNA one-time-pad encryption schemes: (i) a substitution method using libraries of distinct pads, each of which defines a specific, randomly generated, pair-wise mapping; and (ii) an XOR scheme utilizing molecular computation and indexed, random key strings. These methods can be applied either for the encryption of (appropriately recoded) natural DNA, and also for the encryption of DNA encoding binary data. In the latter case, we also present a novel use of chip-based DNA micro-array technology for 2D data input and output. Finally, we examine a class of DNA steganography systems, which secretly tag the input DNA and then disguise it (without further modifications) within collections of other DNA. We consider potential limitations of these steganography methods, showing that with some assumptions on the information theoretic entropy of the plaintext messages, certain DNA steganography systems may not be cryptographically secure, and can be broken. We also discuss various modified DNA steganography systems which appear to have improved security.

1. Introduction

1.1. Biomolecular Computation. Biotechnological methods (e.g., recombinant DNA) have been developed for a wide class of operations on DNA and RNA strands, including site-specific edits and splicing operations. There has recently arisen a new area of research known as DNA computing, which makes use of recombinant DNA techniques for doing computation. More generally, biomolecular computing (BMC) makes use of such biotechnological methods for doing computation (see the comprehensive survey of Reif [R98]). Various recombinant DNA operations have been shown sufficient to allow for universal computation [H92]. BMC methods have been proposed to solve difficult combinatorial search problems such as the Hamiltonian path problem [A94] and the Data Encryption Standard (DES) (see Boneh, et al [BDL95] and Adleman, et al [ARRW96]), using the vast parallelism available via recombinant DNA to do the combinatorial search among a large number of possible solutions represented by DNA strands. While these methods for solving hard combinatorial search problems may succeed for fixed sized problems, they are ultimately limited by their volume requirements, which may

2000 *Mathematics Subject Classification.* Primary 92C40, 92B05, 68Q22, 68Q25, 94A60.

Preprint of paper appearing in "Aspects of Molecular Computing - Essays dedicated to Tom Head on the occasion of his 70th Birthday", Springer Verlag series in Natural Computing (edited by N. Jonoska, G. Paun and G. Rozenberg) LNCS 2950 Festschrift, Springer, pp. 167-188, (2004).

grow exponentially with input size. However, BMC has many exciting further applications beyond pure combinatorial search. For example, DNA and RNA are appealing media for data storage due to the very large amounts of data that can be stored in compact volume. They vastly exceed the storage capacities of conventional electronic, magnetic, optical media. A gram of DNA contains about 1021 DNA bases, or about 108 tera-bytes. Hence, a few grams of DNA may have the potential of storing all the data stored in the world. Most recombinant DNA techniques can be applied at concentrations of 5 grams of DNA per one liter of water. The DNA strands may constitute (a) a "wet" data base of biological data, or (b) data obtained from more conventional binary storage media. In the former case (a), the natural DNA obtained from biological sources may be recoded using nonstandard bases as described in Landweber and Lipton [LL97], to allow for subsequent BMC processing. In the later case (b), the DNA data may be moved to conventional binary storage media via a variety of techniques including DNA chip arrays (see Section 3), and the binary data may be encoded in DNA strands by use of an alphabet of short oligonucleotide sequences. Baum [B95] has discussed methods for fast associative searches within DNA databases using hybridization. Other BMC techniques (see Reif [R95]) might perform more sophisticated data base operations on DNA data such as the data base join operations and various massively parallel operations on the DNA data.

1.2. Cryptography. Data security and cryptography are critical aspects of conventional computing and may be also important to possible DNA data base applications. Here we provide basic terminology used in cryptography (see Schneier [S96]). Suppose a sender wishes to send a message to a receiver so that no one else can read the message. The message is initially written in plaintext (i.e., in non-encrypted form) over a finite alphabet. Encryption (also known as encipherment) is the process of scrambling the plaintext message, transforming it into an encrypted message known as cipher text. For example, a fixed codebook may be used to provide an initial mapping from characters in the finite plaintext alphabet to a finite alphabet of codewords, and then a sophisticated algorithm depending on a key may be applied to further encrypt the message. Decryption (also known as decipherment) is the reverse process of transforming the encrypted message back to the original plaintext message. A cryptosystem (also known as a cipher) is a method for both encryption and decryption of data. Hence, the processes of encryption and decryption require: an informational message to be transmitted; a physical system for message read-in, transformation, and read-out; and agreed upon encryption and decryption algorithms. An unbreakable cryptosystem is one for which successful cryptanalysis is not possible.

1.3. Our Results: BMC methods for Cryptography. This paper investigates a variety of biomolecular methods for encrypting and decrypting data. We assume the plaintext message data is encoded in DNA strands. For example, the DNA strands in solution in a test tube may form a "wet" data base of biological data (e.g., the DNA of personnel of an organization) which might need to be secured (this secrecy might be needed for privacy reasons, or perhaps in the context of our current administration, for legal or political reasons). Alternatively, a test tube of DNA might contain encoded data obtained from more conventional binary or analog storage media. We briefly discuss input and output of the DNA data to conventional binary storage media via (photo- sensitive and/or photo-emitting) DNA chip arrays. In this case we assume the plaintext message data is encoded

in DNA strands by use of a (publicly known) alphabet of short oligonucleotide sequences. In this latter case, one important potential advantage of using DNA for storage is its compactness. First we present a class of DNA cryptography techniques that are in principle unbreakable. We initially secretly assemble a library of one-time-pads in the form of DNA strands. Then we propose a number of methods whereby a large number of short message sequences can be encrypted using these one-time-pads. The aspect that restricts use one-time-pad encryption systems in practice for conventional electronic media is the large amount of one-time-pad data which must be created and transmitted securely. Here, DNA shines because it is an extremely compact form of representing data and significant amounts of information can be carried in a limited amount of physical space. Further, we present an interesting concrete example of a DNA cryptosystem technique in which the input is a two-dimensional image. By use of photo- sensitive DNA-on-a-chip technology, this image is converted to a solution of DNA strands encoding the image, these strands can then be encrypted via DNA cryptography techniques. The reverse process of decryption of the image is also described. The decrypted image can be output as a two-dimensional image, by conventional (fluorescent) DNA-on-a-chip technology. Finally, we discuss a general class of methods, known as steganography, where the original plaintext is not actually encrypted but instead disguised within other data. In DNA steganography methods, the DNA plaintext messages are appended with one or more secret keys and the resulting appended DNA strands are then hidden by mixing them within many other irrelevant DNA strands (e.g., randomly constructed DNA strands). For example, Clelland, Risca, and Bancroft [CRB99] recently proposed genomic steganography: techniques using amplifiable microdots. As in conventional steganography methods, the plaintext messages are not otherwise preprocessed, and so the methods are very appealing due to their simplicity. Unfortunately, we show, with some assumptions on the bounded information theoretic entropy of the plaintext messages found in practice, that a class of such DNA steganography systems can be broken. We also discuss some improvements that increase the security of DNA steganography systems.

1.4. Organization. In this Section 1, we have introduced BMC and cryptography terminology, and also have discussed our results. In Section 2, we describe some unbreakable DNA crptosystems using randomly assembled one-time pads. In Section 3, we give a concrete, detailed example of a DNA cryptosystem for two dimensional images, using a DNA chip for I/O and also using a randomly assembled one-time pad. In Section 4, we discuss a class of DNA steganography techniques and show that they can be broken with some modest assumptions on the entropy of the plaintext, even if they employ perfectly random one-time pads. We conclude our paper in Section 5.

2. DNA Cryptosystems Using Random One-Time Pads

The input plaintext messages. As mentioned above, we assume input to our DNA cyptosystems are short segments of distinct plaintext messages. See below for a discussion of the design of word libraries and see Section 3 for details of a possible method (a chip-based DNA micro-array) for input and output of messages from conventional storage media. One-time-pad encryption uses a random codebook to convert short segments of plaintext messages to encrypted text. Two points critical to security issues should be made about codebooks; they must be truly random, and they must be used only once. This class of cryptosystems using a secret random one-time pad are the only cryptosystems known to be absolutely unbreakable; see

Schneier[S96]. (In contrast, the security of encryption using pseudo-random encryption pads is uncertain; while certain such systems have been shown to be attractive targets for cryptanalysis since their inherent periodicity supplies information to potential code breakers, Yao [Y82] has proven other systems to be unbreakable in polynomial time, given some (as yet unproven) certain complexity theoretic assumptions.) The reuse of pads, likewise, increases the risk of message decryption by an undesired interceptor. These two principles dictate certain features of the DNA sequences and of sequence libraries, which will be discussed more below. We will first assemble a large one-time-pad in the form of a DNA strand, which is randomly assembled from short oligonucleotide sequences, and then isolated and cloned. These one-time-pads will be assumed to be constructed in secret, and we further assume that specific one-time-pads are shared in advance by both the sender and receiver of the secret message. This assumption requires initial communication of the one-time-pad between sender and receiver, which is facilitated by the very compact nature of DNA in solution. We will provide methods for constructing such a DNA one-time-pad. We propose two methods whereby a large number of short message sequences can be encrypted, to ensure the original plaintext message cannot be determined from the resulting DNA. In these encryption methods, the one-time-pad is used in effect as a random codebook (as opposed to a fixed codebook, which cannot be assumed to be secret in an unrestricted cryptosystem) for mapping short message sequences into encrypted sequences. The mapping methods used by these encryption methods include: (1) the use of substitution, where we encrypt each message sequence by associated matching with corresponding sections of the one-time DNA pad, or alternatively, (2) the use of bit-wise XOR computation via biomolecular computing techniques. The decryption is done by similar methods.

2.1. Our DNA Cryptosystem Using Substitution. Substitution One-time-pad Cryptosystems. The input to a substitution one-time- pad system, is a plaintext binary message of length n, partitioned into plaintext words of fixed length, and a substitution one-time-pad consisting of a table randomly mapping all possible strings of plaintext words into cipher words of fixed length, such that there is a unique reverse mapping. The plaintext is encrypted by substituting each ith block of the plaintext with the cipher word given by the table, and is decrypted by reversing these substitutions. DNA Implementation. In the case of encryption by substitution, we wish to convert one test tube of short DNA strands (the plaintext messages) into another set of entirely different strands (the encrypted messages) in a random yet reversible way. Our DNA encoded messages are manipulated such that the plaintext are converted to cipher strands and the plaintext strands are removed. Our substitution method requires one-time-pad DNA sequences to accomplish this conversion. The overall scheme involves long DNA pads containing many segments, where each segment contains a cipher word followed by a plaintext word. The cipher word acts as a hybridization site for binding of a primer, which is then specifically appended with a plaintext word to produce word-pairs. The word-pair DNA strands can be used as a type of lookup table in the first step of the conversion of plaintext into cipher text. Additional manipulations, described in later sections of this paper, accomplish the remaining steps to complete encryption and decryption. DNA one-time pads. An ideal library of one-time pads would contain a huge number of pads and each pad would provide a perfectly unique, random mapping of plaintext to cipher word pairs. Our construction procedure

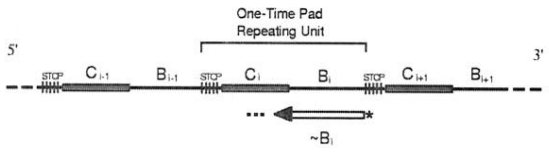

FIGURE 1. One-time-pad Codebook DNA Sequences.

approaches these goals. The structure of an example pad is given in Figure 1. The repeating unit is made up of: one sequence word, Bi, from the set of cipher or codebook-matching words; one sequence word, Ci, from the set of plaintext words; and a polymerase "stopper" sequence. Each sequence pair i, uniquely associates a plaintext word with a cipher word. Oligo with sequence ~Bi, corresponding to the Watson-Crick complement of cipher word Bi, can be used as polymerase primer and be extended by specific attachment of plaintext word Ci. The stopper sequence prohibits extension of the growing DNA strand beyond the boundary of the paired plaintext word. A library of unique codebook strands is constructed using this theme. Each individual strand from this codebook library specifies a particular, unique set of word pairings. The one-time-pad consists of a DNA strand of length n containing $d = n/(L_1 + L_2 + L_3)$ copies of the repeating pattern: a cipher word of length L_2, a plaintext word of length L_1, and stopper sequence of length L_3. (We set $L1 = c_1 \log_2 n, L_2 = c_2 log_2 n$, and $L_3 = c_3$, for fixed integer constants $c_1, c_2, c_3 > 1$.) Each repeat unit specifies a single mapping pair, and no codebook word or plaintext word will be used more than once on any pad. Therefore, given a codebook word B_i we are assured that it maps to only a single plaintext word C_i and vice versa. The stopper sequence acts as "punctuation" between repeat units so that DNA polymerase will not be able to continue copying down the template (pad) strand. Stopper sequences consist of a run of identical nucleotides which act to halt strand copying by DNA polymerase given a lack of complementary nucleotide triphosphate in the test tube. For example, the sequence TTTT will act as a stop point if the polymerization mixture lacks its base-pairing complement, A. Stopper sequences of this variety have been prototyped previously by Hagiya, et al [HA+97]. Given this structure, we can anneal primers and extend with polymerase in order to generate a set of oligonucleotides corresponding to the plaintext/cipher lexical pairings (or word-pair strands). The set of word-pair strands is essentially a lookup table for a random codebook. Therefore, the feasibility depends upon: the size of the lexicon; the number of possible pads available; and the size, complexity, and frequency of message transmissions. Lexicon size d gives the total number of words available for use within a single codebook pad (and, if we use the $2D$ chip grid for I/O (see below Section 3), the number of pixels available on the $2D$ chip grid). Word sizes L_1, L_2 determine the total number of possible DNA sequences available for a lexicon, however lexicon size is reduced by elimination of complementary sequences and division of the possible sequences into two lexicons. Pad diversity stands for the total number of random pads generated during a single pad construction experiment. Multiple construction cycles can be used to prepare greater numbers of unique pads. The estimate of message size relative to lexicon size assumes that the random codebooks are used in conjunction with the DNA chip I/O outlined in Section 3 below.

Codebook Libraries. Construction of the libraries of codebook pads can be approached using segmental assembly or build-up procedures used successfully in previous gene library construction projects [LK93, LB97] and DNA word encoding methods used in DNA computation [DMGFS96, DMGFS98, DMRGF+97, FTCSC97, GDNMF97, GFBCL+96, HGL98, M96]. The design of sequence words for the plaintext and cipher lexicons represents the first technical challenge. We would like the lexicons to be distinct, or disjoint, in the mathematical sense. We also need essentially complete coverage of the lexicon on each pad, as well as unique word mapping, so a single plaintext word pairs with a single cipher word and vice versa. (While perfectly disjoint lexicons are ideal, the system may function with biased lexicons containing a small degree of potential overlap.)

Construction of DNA one-time pads. There are a number of possible methodologies for construction of the plaintext and cipher pairs used for the pad. One methodology is the random assembly of one-time pads in solution (e.g. on a synthesis column). We view such methods less favorably due to the difficulty of achieving both full coverage and yet still avoiding possible conflicts by repetition of plaintext and/or cipher words. (We can set the constants c1 and c2 large enough so that the probability of getting repeated words on a pad of length n is very small, but then the coverage may be reduced.) The methodology, which we favor since it is very controllable, is to employ a DNA chip (see [FRP+91, PSS+94. CYH+96, BKH96. S98] for basic DNA chip technology). Such DNA chips are currently commercially available and chemical methods for construction of custom variants are well developed. The DNA chip has an array of immobilized DNA strands, so that multiple copies of a single sequence are grouped together in a microscopic pixel. The microscopic arrays of the DNA chip are optically addressable, and there is known technology for growing distinct DNA sequences at each (optically addressable) site of the array. Light-directed synthesis allows the chemistry of DNA synthesis to be conducted in parallel at thousands of locations (i.e. it is a combinatorial synthesis). Therefore, the number of sequences prepared far exceeds the number of chemical reactions required. For preparation of oligonucleotides of length L, the 4L sequences are synthesized in 4n chemical reactions. For example, the ~65,000 sequences of length 8 require 32 synthesis cycles, and the 1.67x107 sequences of length 10 require only 48 cycles. The plaintext and cipher pairs can be constructed so that there is nearly complete coverage of the lexicon on each pad, as well as a nearly unique word mapping between plaintext and cipher pairs. These resulting cipher word, plaintext word pairs can be assembled together in random order (with possible repetitions) on a long DNA strand by a number of known methods (e.g., simply blunt end ligation, or for higher efficiency, the use of a technique of hybridization assembly with complemented pairs as done in Adleman's original DNA experiment [A97]). Cloning or PCR may be used to amplify the resulting one-time pad.

2.2. Our DNA XOR One-time-pad Cryptosystem. *Vernam Cipher Cryptosystems:* In classical cryptography (see Kahn[K67]), the Vernam cipher (now known as the XOR one-time-pad cryptosystem) is deployed by generating a sequence, S, of R independently distributed random bits known as a one-time-pad, replicating the one-time-pad, and storing one copy at the source and one at the destination. Let L be the number of bits of S that remain unused, where initially $L = R$. Recall that XOR is the operation that given two Boolean inputs, yields 0 if the inputs are the same, and otherwise is 1. When a plaintext binary message M which is $n < L$ bits long needs to be sent, each bit M_i is XOR'ed with the

bit $K_i = SR - L + i$ to produce encrypted bits $Ci = M_i \text{XOR} K_i$ for $i = 1, \ldots, n$. The n bits of S that have been consumed are then destroyed at the source and the encrypted sequence $C = (C_1, C_2, ..., C_n)$ is despatched to the destination. At the destination the identical process is repeated - that is the sequence C is used in the place of M, performing bitwise XOR with bits from S, destroying the bits of S after they are consumed. The commutative property of the XOR results in the initial message being reproduced since $C_i XOR K_i = M_i$ since $C_i = M_i XOR K_i$ and Mi XOR K_i XOR$K_i = M_i$. (Note that for efficient DNA encoding, we may alternatively do this in quaternary, that is over modular base 4, rather than in binary, since DNA has four nucleotides. In this case, we assume the input plaintext and one-time-pad are in quaternary, and for encryption we use addition modulo 4 instead of XOR, and for decryption we subtract the one-time-pad elements modulo 4. For simplicity, we will continue the discussion for the binary case.) We wish again to convert one test tube of short DNA strands (the plaintext messages) into another set of entirely different strands (the encrypted messages) in a random yet reversible way. Each of the plaintext messages are assume to have appended unique prefix index tags of fixed length L0 indexing them. Each of the one-time-pad DNA sequences are also assumed to have appended unique prefix index tags of the same length L0, which form the complements of the plaintext message tags. By use of known recombinant DNA techniques (e.g., annealing and ligation), each corresponding pair of a plaintext message and a one-time-pad sequence, with the same tag, can be concatenated into a single DNA strand. Our DNA encoded messages are modified in this case by a bit-wise XOR computation, so that fragments of the plaintext are converted to cipher strands using the one-time-pad DNA sequences, and the plaintext strands are removed. The reverse decryption is similar, using the commutative property of the bit-wise XOR operation. Clearly, as the remaining basic building block, we need to describe a method to effect bit-wise XOR on vectors. Methods developed to effect binary arithmetic operations such as addition are similar to the bit-wise XOR computation, but have the additional constraint of needing to deal with a carry bit. If we have a method to execute addition, we could use it, ignoring the carry bit and slightly altering the mapping to reflect the rules of bit-wise XOR instead of those of bit addition (which takes into account also carry bits). The required bit-wise XOR computation can thus be done by such modification (which involves a considerable simplification) of at least two previously known biomolecular computing techniques for integer addition:

(a) Guarnieri, Fliss, and Bancroft prototyped [GFB96] the first BMC addition operations (on single bits) in recombinant DNA. This experimental work was very significant. However, it suffered from some limitations: (i) only two numbers where added, so it did not take advantage of the massive parallel processing capabilities of BMC and (ii) the outputs were encoded distinctly from the inputs, so it did not allow for repeated operations. Subsequent proposed methods [OGB97,LKSR97,GPZ97] for basic operations such as arithmetic (addition and subtraction) permit chaining of the output of these operations into the inputs to further operations, and to allow operations to be executed in massive parallel fashion. In particular, Rubin el al [RKL98] gave an experimental demonstration of a BMC method for chained integer arithmetic. This work also gave one of the first demonstrations in BMC of logically reversible computation, by way of the conditional XOR operation. This method, which O(n) recombinant DNA operations for

FIGURE 2. Simulated Read-Outs from DNA Chip Input/Output.

vectors of lengh n, can be used directly for our required XOR vector operations (i.e., by simply disabling the logic used for carry-sums from previous bits).

(b) Another method being developed for effecting general binary addition, and XOR in particular as well, is by the use of DNA tiles (LaBean, et al [LWR99]). Their idea is based on the DNA self assembly work by Winfree, et al [W95, W96, Win98a, WLW+98, WYS96] and the compact assembly methods of Reif [R97]. Tiles with specific uncomplemented pads at the corners were constructed, with the purpose of effecting self-assembly. The tiles that have been prototyped in the laboratory by LaBean have been designed so that self-assembly of an output of a computation can occur based on an initial assembly created from tiles that represent the input. More specifically, given a binary input string, each bit is represented by a single tile. The tiles are designed such that they assemble linearly to represent the binary string. The use of a special corner tile allows two such linear tile assemblies representing two binary input strings respectively, to come together and create a closed framework within which specially designed output tiles can fit in. This process allows for unmediated parallelized binary addition or XORing. At the end of the process, there exists a single strand that runs through the entire assembly which will contain the two inputs and the output, as a result of the special design of these tiles. By using this property we are able to effect the Vernam cipher in DNA. For details of this XOR computation, using self-assembly of DNA tiles, see LaBean, et al [LYR+98,LWR99].

3. A DNA Cryptosystem for 2D Images using a DNA Chip and a Randomly Assembled One-Time Pad

3.1. Overview of Method. In this section, we outline an example system capable of encryption and decryption of input and output data in the form of 2D images recorded on the microscopic arrays of a DNA chip. (Again, see [FRP+91, PSS+94. CYH+96, BKH96. S98] for basic DNA chip technology and also see Mills, et al [MYP98] for a previous use of DNA chips for I/O.) The system we describe here consists of: a data set to be encrypted; a chip bearing immobilized DNA strands; and a library of one-time pads encoded on long DNA strands. The data set for encryption in this specific example is a 2-dimensional image, but variations on the method may be useful for encoding and encrypting other forms of data or types of information. Recall the DNA chip contains an addressable array of nucleotide sequences immobilized such that multiple copies of a single sequence are grouped together in a microscopic pixel. Again, we note that such DNA chips are currently commercially available and chemical methods for construction of custom variants are well developed. Further chip details will be given below. The library of one-time pads was described in Section 2.1 above. Figure 2 gives a coarse grained outline of the I/O method. Fluorescent-labeled, word-pair DNA strands are prepared from a substitution pad codebook as described in Section 2.1. These are annealed specifically to their sequence complements at unique sites (pixels) on

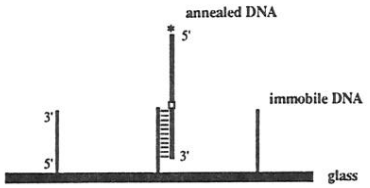

FIGURE 3. Components and Organization of the DNA Chip.

the DNA chip. The message information (Panel A) is transferred to a photo mask with transparent (white) and opaque (black) regions. Following a light-flash of the mask-protected chip, the annealed oligonucleotides beneath the transparent mask pixels are cleaved at a photo-labile position; and their 5′ sections dissociated from the annealed 3′ section and are collected in solution. This test tube of strands is the encrypted message. Annealed oligos beneath opaque mask are unaffected by the light-flash and can be subsequently washed off the chip and discarded. If the encrypted message oligos are reannealed onto a (washed) DNA chip, they would now anneal to new specifc locations and the message information would be unreadable (Panel B). (Note: if the sequence lexicons for 5′ segment (cipher word) and 3′ segment (plaintext word) are disjoint, no binding would occur and the chip in Panel B would be completely black.) Decrypting is accomplished by using the fluorescent labeled oligos as primers in one-way (lopsided) PCR with the same one-time codebook which was used to prepare the initial word-pair oligos. When the word-pair PCR product is bound to the same DNA chip, the decrypted message is revealed (Panel C). Input/Output as observed by fluorescence microscopy of the DNA I/O chip. presents a coarse grained outline of the I/O method. The annealed DNA in Figure 3 corresponds to the word-pair strands prepared from a random substitution pad as described in Section 2.1 above. Immobile DNA strands are located on the glass substrate of the chip in a sequence addressable grid according to currently used techniques. The annealed strand contains: a fluorescent label on its 5′ end (asterisk); a codebook-matching sequence word (not base-paired on the chip); a photo-labile base (white square) capable of cleaving the DNA backbone; and a chip-matching word (base-paired to immobile strand). The 5′ (unannealed) end carries a cipher word while the 3' (annealed) end carries a plaintext word. These word-pair strands are special in that they contain a photo-cleavable base analog between the two sequence words. The photo-cleavable base analog can be chemically added to the 3′ end of the cipher word during oligo synthesis. Figure 4 gives step by step procedures for encryption and decryption. We start with a DNA chip displaying the sequences complementary (in the Watson-Crick sense) to the plaintext lexicon. In step one, the fluorescent-labeled word-pair strands prepared from a one-time-pad are annealed to the chip at the pixel bearing the complement to their plaintext 3′ end. In the next step, the mask (heavy black bar) protects some pixels from a light-flash. At unprotected regions, the DNA backbone is cleaved at a site between the plaintext and cipher words. In the final step, the cipher word strands, still labeled with fluorophore at their 5′ ends, are collected and transmitted as the encrypted message. A message can be decrypted only by using a one-time-pad and DNA chip identical to those used in the encrypting process. First, the word-pair strands must be reconstructed by appending the cipher word with the proper plaintext word. This is accomplished by polymerase extension or lop-sided PCR using cipher words as primer and one-time-pad as template. The cipher strands bind to their specific locations on the pad and are appended with their proper plaintext

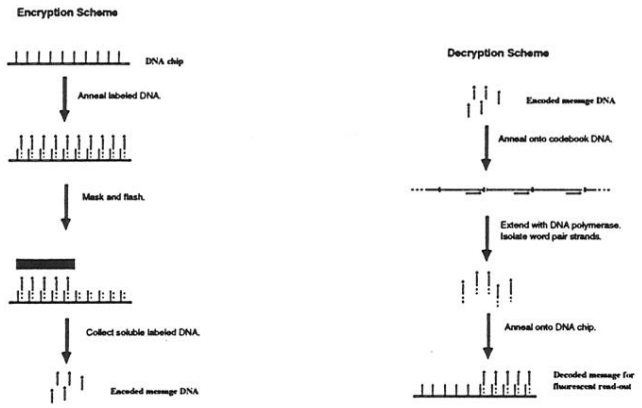

FIGURE 4. Step by step procedures for encryption and decryption.

partner. Note that in the decrypting process the fluorescent label is still required, but the photo-labile base is unnecessary and not present. The final step of decryption involves binding the reformed word-pair strands to the DNA chip and reading the message by fluorescent microscopy.

4. DNA Steganography Techniques

4.1. Steganography. Steganography is a class of techniques that hide secret messages within other messages. In a steganography system, the original plaintext is not actually encrypted but is instead disguised or hidden within other data. Historical examples of steganography systems are the use of grills that mask out all of an image except the secret message, micro-photographs placed within larger images, invisible inks, etc. The cryptography literature (see Schneier[S96]) generally consider conventional steganography methods to have low security (in fact there is disagreement whether steganography is actually encryption, since the plaintext is not actually encrypted but instead disguised within other media) and there are numerous cases where steganography methods have been broken in practice (e.g., see Kahn[K67] and Schneier[S96]). However, it is very appealing due to it's simplicity. There are a number of techniques for applying steganography in the context of biomolecular computation. One method is to take one or more input DNA strands (considered to be the plaintext message) and append to them one or more randomly constructed "secret key" strands. The resulting "tagged plaintext" DNA strands are then hidden by mixing them within many other additional "distracter" DNA strands which might also be constructed by random assembly. Given knowledge of the "secret key" strands, the resulting solution of DNA strands can be decrypted by a number of possible known recombinant DNA separation methods. For example, the plaintext message strands may be separated out by hybridization with the complements of the "secret key" strands might be placed in solid support on magnetic beads or on a prepared surface. These separation steps may combined with amplification steps and/or PCR (see Barnes [B94] and Roberts [R94]).

4.2. Cryptanalysis of DNA Steganography Systems. In a true cryptographic cipher the security is dependent on a secret key. The security of the above system is expected to derive from – as is the case with all steganographic systems – the fact that the adversary is unaware of the existence of the message in the medium of transmission, and/or can not distinguish the plaintext message from

the medium. As soon as this assumption is no longer valid, the system can generally be compromised. In particular, this DNA steganography system's security is entirely dependent on the degree that the message DNA strands are indistinguishable from the "distracter" DNA strands. It is reasonable to assume the secret tags are indistinguishable from the "distracter" DNA strands. However, if (a) the plaintext is not initially compressed, and comes from a natural source such as English or natural DNA, and (b) the "distracter" DNA strands are constructed by random assembly, then the original plaintext portion of the "tagged plaintext" DNA strands are distinguishable from the other additional "distracter" DNA strands. We now consider cryptanalysis of such a steganography system, without knowledge of the "secret key" strands, and with the assumption that the plaintext is generated from a source that is distinguishable from the additional "distracter" DNA strands that are mixed with them. In particular, we assume that we know both (i) the probability distribution of the source generating the set of plaintext message DNA strands and we also know (ii) the probability distribution of the source generating the set of "distracter" DNA strands, and we further assume these probability distributions are distinct. The Shannon information theoretic entropy E_S (see [CT 91]) provides a measure of the factor that a source can be compressed without loss of information. For example, many images have entropy nearly 4, English text has entropy about 3, and computer programs have entropy about 5. The overall entropy of most classes of DNA, (including human DNA) is in the range of 1.2 to 2 (see [GT94], [LY97],[NW99]), and is larger on non-coding sequences than on the protein coding sequences. As a simple example, let us for the moment assume: (i) the plaintext DNA source has Shannon entropy E_S bounded by a constant above 1, (ii) the probability distribution of the plaintext source is known (e.g., generated by an ergodic stationary process), and (iii) the "distracter" DNA strands have a random uniform distribution. We also assume for simplicity that both the plaintext and "distractor" DNA strands are of some large length n. Let $L = E_S \log_2 n$. Then, following certain known lossless data compression schemes such as Lempel-Ziv [ZL77] (also see [CT 91]), we form a dictionary D of the $d = n/L$ most frequently occurring subsequences of length at least L in the known plaintext source distribution. We will separate out the plaintext message strands by repeated rounds of hybridization with the complements of the elements of this dictionary. On each round of separation, we assume a test tube T which contains a mixture of "tagged plaintext" DNA strands mixed with a high concentration of "distracter" DNA strands. Let $r(T)$ be the ratio of concentration of "distracter" DNA strands to "tagged plaintext" DNA strands. We can assume that we have constructed a dictionary D whose size is is much larger than $\log r(T)$. On each round we form a new test tube $F(T)$ with expected $r(F(T))$ considerably reduced from the previous ratio $r(T)$, by the following steps:

[1] Pour a fraction s (e..g, let $s = 1/2$) of the volume of the current test tube T into a test tube T_1 and pour the remaining fraction $1 - s$ of the volume T into a test tube T_2.

[2] Choose a random text phrase x in D (not previously considered in a prior trial), and using the Watson-Crick complement of that phrase x, do a separation on test tube T_2 with respect to phrase x, yielding a new test tube T_3 whose contents are only DNA strands containing the phrase x.

[3] Pour the contents of test tubes T_1 and T_3 into a new test tube $F(T)$.

(Note that the series of separations, used to eliminate the distracters, will shape the distracter population into one whose distribution more closely matches that of the plaintext. Nevertheless, our proof given below will remain correct on subsequent trails, since on each trail we choose a random text phrase from the dictionary D that had not previously been considered in a prior trial. Furthermore, the distracters are assumed to have a random uniform distribution. Hence we can assume independent probabilities from trial to trial.) It will be useful to note that for large n, we can approximate $(1-1/n)^n \sim 1/e$. By known results used in lossless data compression methods (see Cover and Thomas [CT 91]), each subsequence of length L of the plaintext DNA strand whose source was the same as used in formation of the dictionary D, the probability of containing a given text phrase x of length at least L from the dictionary D is expected to be $1/d$. Since the plaintext DNA strand is assumed to be of length n, and contains at least d= n/L distinct phrases of length L, it follows that it has probability at most $(1-1/d)^d \sim 1/e$ of not containing dictionary text phrase x. (Note that since this event is not a certainty, the above protocal uses re- pooling of test tubes T_1 and T_3.) On the other hand, each subsequence of length L of a "distracter" DNA strand is random, so has probability $p = 1/4^L = 1/4^{(\log n)E_S} = 1/n^{2E_s}$ of containing a given text phrase of length at least L. Since the "distracter" DNA strands are assumed to be of length n, it follows that each of them has probability at least $(1-p)^n$ of not containing a given text phrase of length at least L. But this probability is $(1-p)^n = (1-p)^{(1/p)(np)}$ $\sim (1/e)^{np} = 1/exp(np) = 1/exp(n^{1-2E_S}) \sim 1$ for large n and entropy $E_S > 1$. Therefor, we have shown that it is very unlikely that the "distracter" DNA strands contain a given random text phrase x of length at least L from the dictionary D. Hence, we have shown: (1) with probability at most $1 - 1/exp(n^{1-2E_S}) \sim 0$, any given "distracter" DNA strand originally within test tube T_2 is added to test tube T_3 via the separation step, and (2) in contrast, with probability at least $1-1/e$, any plaintext DNA strand originally within test tube T_2 is added to test tube T_3 via the separation step. Since test tubes T_1, T_2 consist of fractions $s, 1-s$ of the volume test tube T, and $F(T)$ is constructed by pouring together test tubes T_1, T_3 we have that the relative concentration of DNA strands in $F(T)$ as compared to the original test tube T expects to decrease by: (1') a factor of $s + (1-s)/exp(n^{1-2E_S}) \sim s$ for the "distracter" DNA strands, and (2') a factor of $s + (1-s)/e$ for any plaintext DNA. Let $c = 1/(1-1/e+1/s) < 1$. It follows that the ratio $r(F(T))$ of "distracter" DNA strands to plaintext DNA will expect to decrease from the original ratio $r(T)$ by at least this constant multiplier c factor: $(s + (1-s)/exp(n^{1-2E_S}))/(s + (1-s)/e)$ $\sim s/(s + (1-s)/e) = 1/(1 + (1/s - 1)/e) = c < 1$, for $0 < s < 1$. Hence, for any given $r' < r$, after $\log(r/r')/\log(1/c) = O(\log(r/r'))$ repeated rounds of this process, the ratio of concentration in the test tube T will expect to decrease from initially $r = r(T)$ to the given smaller ratio r'. Cryptanalysis using hints. Another cryptanalysis technique for breaking steganographic systems is to use a number of "hints" that disambiguate the plaintext. As a concrete example, consider the case where we wish to make secret the DNA of an individual (say, the President), and we do this using such an improved steganography system where the other additional "distracter" DNA strands (that are mixed with the DNA of an individual) are DNA from a similar but not identical genetic pool. Then the steganography system may often be broken by use of sufficiently many distinguishing "hints" concerning the DNA of the individual, such as the fact that the individual might have a particular set of observable expressed gene sequences (e.g., for baldness, etc.). These hints

may allow for the subsequent identification of the full secret DNA by use of a series of separation steps using the complement of portions of the known gene sequences.

4.3. Improved DNA Steganography Systems. We may enhance the security of DNA steganography systems by a number of methods:

A. Mimicking Distribution of "Distracter" DNA. The first technique is an improved construction of the set of "distracter" DNA strands, so that their distribution better mimics the plaintext source distribution. In particular, we can construct the "distracter" DNA strands by random assembly from elements of the dictionary D, so it is more difficult to distinguish the probability distribution of the plaintext source from that of the "distracter" DNA strands. (Also note that we may alternatively apply a series of separations, used to eliminate the distracters in our above cyptoanalysis of steganography systems, to shape the distracter population into one whose distribution more closely matches that of the plaintext.) This difference between these two distributions (that is, between the source distribution and the distribution resulting from random assembly from elements of the dictionary D) can be formally given by the relative entropy (see definition in Cover and Thomas [CT 91]). It is possible that such an improved steganography system may be in turn susceptible to a modified cryptanalysis attack - there may remain enough special properties of realistic source distributions such as English text and natural DNA that are not reflected in the fixed size dictionary D, so that the resulting relative entropy might still be sufficient to break the resulting DNA steganographic system. In particular, the resulting enhanced DNA steganographic system might still be broken by simply using a similar cryptanalysis technique as given above, but with instead a somewhat larger dictionary D', providing a better model of the source plaintext distribution.

B. Compression of the Plaintext. Another method for an improved DNA steganography system is to recode the plaintext using a universal lossless compression algorithm method such as Lempel-Ziv [ZL77]. In this case the resulting distribution of the recoded plaintext approximates a universal distribution, so uniformly random assembled distracter sequences may suffice to provide improved security. The drawback of this system is that unlike conventional steganography methods, the plaintext messages needs to be preprocessed. This recoding of plaintext messages can easily be done electronically; but the recoding would have to be done by BMC means in the case where the plaintext is derived from natural DNA. In the latter case, we can do the recoding by a method similar to that discussed in Section 2 for substitution one-time-pads (using a Lempel-Ziv dictionary D for the substitution mapping). There is no known proof of the security of the resulting enhanced DNA steganography system; in contrast to the absolute security provided by the use of one-time-pads for equivalent effort.

5. Conclusion

This paper has presented an initial investigation into the use of DNA-based methods for cryptosystems. We have discussed two classes of methods: (i) DNA cyptography methods based on DNA one-time-pads, and (ii) DNA steganography methods. Our DNA substitution and XOR methods based on one-time-pads are in principle unbreakable. Furthermore, we have provided an example of a concrete implementation of our DNA cyptography methods which includes 2D input and output. In contrast, it remains to be seen if DNA steganography systems with natural DNA plaintext input can or cannot be made to be unbreakable. We have

shown that a certain class of DNA steganography methods offer only limited security, and can be broken with some reasonable assumptions on the entropy of the plaintext messages. We also considered a number of modified DNA steganography systems with apparently improved security.

Acknowledgements. The authors wish to sincerely thank the referees for their excellent suggestions for improvements to this paper, and also wish to thank Jeremy De Bonet for helpful editorial comments. This work was supported in part by Grants NSF/DARPA CCR-9725021, CCR-96-33567, NSF IRI- 9619647, ARO contract DAAH-04-96-1-0448, and ONR contract N00014-99-1-0406.

References

[A94] Adleman, L., Molecular Computation of Solution to Combinatorial Problems, Science, 266, 1021, (1994).

[A95] Adleman, L., On Constructing a Molecular Computer, Dept of CS, U.S.C., (1995). Available via anonymous ftp from
ftp.usc.edu/pub/csinfo/papers/adleman/molecular_computer.ps.

[ARRW96] Adleman, L.M., P.W.K. Rothemund, S. Roweis, E. Winfree, On Applying Molecular Computation To The Data Encryption Standard, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton, June, 1996

[BCGT96] Bach, E., A. Condon, E. Glaser, and C. Tanguay, Improved Models and Algorithms for DNA Computation, Proc. 11th Annual IEEE Conference on Computational Complexity, J. Computer and System Sciences, to appear.

[B94] Barnes, W.M., PCR amplification of up to 35-kb DNA with high fidelity and high yield from bacteriophage templates, Proc. Natl. Acad. Sci., 91, 2216–2220, (1994).

[B95] Baum, E. B., How to build an associative memory vastly larger than the brain, Science, April 28, 1995.

[B96] Baum, E. B., DNA Sequences Useful for Computation, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, June 1996.

[BKH96] Blanchard, A. P., R. J. Kaiser and L. E. Hood, High-density oligonucleotide arrays, Biosens. Bioelec., Vol. 11, 687-690, (1996).

[BDL95] Boneh, D., C. Dunworth, R. Lipton, Breaking DES Using a Molecular Computer, Princeton CS Tech-Report number CS-TR-489-95, (1995).

[CYH+96] Chee, M. ,R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris and S. P. A. Fodor, Accessing genetic information with high-density DNA arrays, Science, Vol. 274, 610-614, (1996).

[CT 91] Cover, T. M. and J. A. Thomas, Elements of Information Theory, John Wiley, NY, (1991).

[DMGFS96] Deaton, R., R.C. Murphy, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., Good encodings for DNA-based solutions to combinatorial problems, Proceedings of the 2nd Annual DIMACS Meeting on DNA Based Computers, June 1996.

[DMGFS98] Deaton, R., R.C. Murphy, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., Reliability and efficiency of a DNA-based computation, Phys. Rev. Lett. 80, 417-420 (1998).

[DMRGF+97] Deaton, R., R.C. Murphy, J.A. Rose, M. Garzon, D.R. Franceschetti, and S.E. Stevens, Jr., A DNA Based Implementation of an Evolutionary Search for Good Encodings for DNA Computation, ICEC'97 Special Session on DNA Based Computation, Indiana, April, 1997.

[CRB99] Clelland, C.T., Risca, V., and C. Bancroft. Genomic Steganography: Amplifiable Microdots. To appear in Nature, 1999.

[FRP+91] Fodor, S. P. A., J. L. Read, C. Pirrung, L. Stryer, A. T. Lu and D. Solas, Light-directed spatially addressable parallel chemical synthesis, Science, Vol. 251, 767-773, (1991).

[FTCSC97] Frutos, A.G., A.J. Thiel, A.E. Condon, L.M. Smith, R.M. Corn, DNA Computing at Surfaces: 4 Base Mismatch Word Design, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).
[GDNMF97] Garzon, M., R. Deaton, P. Neathery, R.C. Murphy, D.R. Franceschetti, S.E. Stevens Jr., On the Encoding Problem for DNA Computing, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).
[GFBCL+96] Gray, J. M. T. G. Frutos, A.M. Berman, A.E. Condon, M.G. Lagally, L.M. Smith, R.M. Corn, Reducing Errors in DNA Computing by Appropriate Word Design, University of Wisconsin, Department of Chemistry, October 9, 1996.
[GT94] Grumbach, S., and F. Tahi, Compression of DNA Sequences, Proceedings of the IEEE Data Compression Conference (DCC'94), Snowbird, UT, 72–82, March 1994.
[GB96] Guarnieri, F., and C. Bancroft, Use of a Horizontal Chain Reaction for DNA-Based Addition, Proceedings of the 2nd Annual DIMACS Meeting on DNA Based Computers, June 10-12, 1996, American Mathematical Society, Providence, RI (in press), (1996).
[GFB96] Guarnieri, F., Fliss, M., and C. Bancroft, Making DNA Add, Science, 273, 220–223, (1996).
[GPZ97] Gupta, V., S. Parthasarathy, M.J. Zaki, Arithmetic and Logic Operations with DNA, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).
[HGL98] Hartemink, A., D. Gifford, J. Khodor, Automated constraint-based nucleotide sequence selection for DNA computation, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (1998).
[HA+97] Hagiya, M., M. Arita, D. Kiga, K. Sakamoto, and S. Yokoyama, Towards Parallel Evaluation and Learning of Boolean mu-Formulas with Molecules, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997).
[H92] Head, T., Splicing schemes and DNA, In: Lindenmayer Systems: Impacts on Theoretical Computer Science, Computer Graphics, and Developmental Biology, Ed. by G.Rozenberg and A.Salomaa, Springer-Verlag, 371-383, (1992). Also appears in: Nanobiology 1, 335–342, (1992).
[HH92] Henikoff, S. and Henikoff, J.G, Amino acid substitution matrices from protein blocks, Proc Natl Acad Sci, Nov 15;89(22):10915-9 (1992).
[K67] Kahn, D. The Codebreakers: The Story of Secret Writing, New York: Macmillan Pub. Comp., (1967).
[K98] J. Kelsey, B. Schneier, D. Wagner and C. Hall, Cryptoanalytic Attacks on Pseudorandom Number Generators, Lecture Notes in Computer Science, 1998, p 168.
[KBDL98] Kotera,M., A.-G. Bourdat, E. Defrancq and J. Lhomme, A highly efficient synthesis of oligodeoxyribonucleotides containing the 2'-deoxyribonolactone lesion, J. Am. Chem. Soc., 120, 11810-11811, (1998).
[LK93] LaBean, T.H. and S.A. Kauffman, Design of synthetic gene libraries encoding random sequence proteins with desired ensemble characteristics. Protein Science 2, 1249-1254. (1993).
[LB97] LaBean, T.H. and T.R. Butt, (U.S. Patent # 5,656,467), Methods and materials for producing gene libraries. Date of issue 20-August-1997.
[LYR+98] LaBean, T. H., H. Yan, J. H. Reif and N. Seeman, Construction and Analysis of a DNA Triple Crossover Molecule, (November. 1998).
[LWR99] LaBean, T. H., E. Winfree, J. H. Reif, Experimental Progress in Computation by Self-Assembly, 5th Annual DIMACS Meeting on DNA Based Computers, MIT, Cambridge, MA (June 1999).
[LL97] Landweber, L.F. and R. Lipton, DNA 2 DNA Computations: A Potential 'Killer App'?, 3nd Annual DIMACS Meeting on DNA Based Computers, University of Pens., (June 1997).
[LY97] Loewenstern, D. and Yainilos, P., Significantly lower entropy estimates for natural DNA sequences, J.A Storer and M Cohn (Eds.), IEEE Data Compression Conference, Snowbird, UT, pp. 151-161, (March, 1997).

[MYP98] Mills, A., B. Yurke, P. Platzman, Error-tolerant massive DNA neural- network computation, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

[M96] Mir, K.U., A Restricted Genetic Alphabet for DNA Computing, 2nd Annual DIMACS Meeting on DNA Based Computers, Princeton University, (June 1996).

[NW99] Nevill-Manning, C.G. and I.H. Witten, Protein is Incompressible, J.A Storer and M Cohn (Eds.), IEEE Data Compression Conference, Snowbird, UT, pp. 257-266, (March, 1999).

[PSS+94] Pease, A. C. , D. Solas, E. J. Sullivan, M. T. Cronin, C. P. Holmes and S. P. Fodor, Light-generated oligonucleotide arrays for rapid DNA sequence analysis, Proc. Natl Acad. Sci. USA, Vol. 91, 5022-5026, (1994).

[P99] Pirrung, M., personal communication, (1999).

[R95] Reif, J.H., Parallel Molecular Computation: Models and Simulations, Seventh ACM Symp. on Parallel Algorithms and Architectures (SPAA95), ACM, Santa Barbara, 213–223, June 1995. Algorithmica, special issue on Computational Biology, 1999. (http://www.cs.duke.edu/~reif/paper/ paper.html)

[R97] Reif, J.H., Local Parallel Biomolecular Computation, 3rd DIMACS Meeting on DNA Based Computers, Univ. of Penns., (June, 1997). DIMACS Series in Discrete Mathematics and Theoretical Computer Science, ed. H. Rubin, (1999). (http://www.cs.duke.edu/~reif/paper/Assembly.ps and /Assembly.fig.ps)

[R98] Reif, J.H., Paradigms for Biomolecular Computation, First International Conference on Unconventional Models of Computation, Auckland, New Zealand, January 1998. Unconventional Models of Computation, edited by C.S. Calude, J. Casti, and M.J. Dinneen, Springer Pub., Jan. 1998, pp 72-93. (http://www.cs.duke.edu/~reif/paper/paradigm.ps)

[R94] Roberts, S.S., Turbocharged PCR, Jour. of N.I.H. Research, 6, 46–82, (1994).

[RKL98] Rubin, H., J. Klein, T. Leete, A biomolecular implementation of logically reversible computation with minimal energy dissipation, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

[S96] Schneier, B., Applied Cryptography, 2nd Edition, John Wiley, (1996).

[S98] Suyama, A., DNA chips - Integrated Chemical Circuits for DNA Diagnosis and DNA computers, To appear,(1998).

[WQF+98] Wang, L., Q. Liu, A. Frutos, S. Gillmor, A. Thiel, T. Strother, A. Condon, R. Corn, M. Lagally, L. Smith, Surface-based DNA computing operations: DESTROY and READOUT, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

[W95] Winfree, E., Complexity of Restricted and Unrestricted Models of Molecular Computation, Princeton DIMACS Technical Report workshop on DNA-based computers, April 4, 1995.

[W96] Winfree, E., On the computational power of DNA annealing and ligation, DNA based computers, Lipton, R.J. and Baum, E.B. eds., Am. Math. Soc., Providence, RI, (1996).

[Win98a] Winfree, E., Simulations of computing by self-assembly, 4th Int. Meeting on DNA-Based Computing, Baltimore, Penns., (June, 1998).

[WLW+98] E. Winfree, F. Liu, Lisa A. Wenzler, N. C. Seeman, Design and Self- Assembly of Two Dimensional DNA Crystals, Nature 394: 539–544, 1998. (1998).

[WYS96] Winfree, E., X. Yang, N.C. Seeman, Universal Computation via Self- assembly of DNA: Some Theory and Experiments, 2nd DIMACS Meeting on DNA Based Computers, Princeton, June, 1996.

[Y82] Yao, A., Theory and Applications of Trapdoor Functions, Proc. of the 23rd IEEE Symp. On Foundations of Computer Science, pp. 80-91, (1982).

[ZS92] Zhang, Y., and N.C. Seeman, A Solid-Support Methodology for the Construction of Geometrical Objects from DNA, J. Am. Chem. Soc., 114, 2656– 2663, (1992).

[ZL77] J. Ziv and A. Lempel, A Universal Algorithm for Sequential Data Compression, IEEE Transactions on Information Theory, 23:3, 337–343,(1977).

(Ashish Gehani) D223 LSRC Building, Research Drive, Department of Computer Science, Duke University, Box 90129 Durham, NC 27708-0129
E-mail address: geha@cs.duke.edu

(Thomas LaBean) D223 LSRC Building, Research Drive, Department of Computer Science, Duke University, Box 90129 Durham, NC 27708-0129
E-mail address: thl@cs.duke.edu

(John Reif) D223 LSRC Building, Research Drive, Department of Computer Science, Duke University, Box 90129 Durham, NC 27708-0129
E-mail address: reif@cs.duke.edu