DNA AND MOLECULAR COMPUTING

Natasha Jonoska, chair

Algorithmic Self-Assembly of DNA Tiles and its Application to Cryptanalysis

Olivier Pelletier[1]
[1]Accenture Technology Labs
Sophia Antipolis, France
Olivier.Pelletier@accenture.com

André Weimerskirch[2,1]
[2]Electrical Eng. & Information Sciences Dept.
Ruhr-Universität Bochum
Bochum, Germany
weika@crypto.ruhr-uni-bochum.de

Abstract

The early promises of DNA computing to deliver a massively parallel architecture well-suited to computationally hard problems have so far been largely unkept. Indeed, it is probably fair to say that only toy problems have been addressed experimentally. Recent experimental development on algorithmic self-assembly using DNA tiles seem to offer the most promising path toward a potentially useful application of the DNA computing concept. In this paper, we explore new geometries for algorithmic self-assembly, departing from those previously described in the literature. This enables us to carry out mathematical operations like binary multiplication or cyclic convolution product. We then show how to use the latter operation to implement an attack against the well-known public-key crypto system NTRU.

Keywords: DNA, self-assembly, multiplication, convolution product, cryptanalysis, NTRU, Wang tiles

1 Introduction

Since the seminal work of Adleman on the Traveling Salesman Problem (TSP) [2], DNA computing has received a lot of attention both from a theoretical point of view [6] and from an experimental perspective [10, 7]. By using the DNA molecule as a carrier of non-genetic information, and biochemistry as a way to process this information, it is possible to build a massively parallel computing architecture. The implementation details vary from one experimental approach to another, but it is certainly fair to describe the overwhelming majority of the reported experiences in the following way: DNA molecules are used to represent potential solutions and biochemical reactions are used to test whether these solutions satisfy or not the criteria for being an actual solution of the problem. Even though a single biochemical step can take as much as one day to perform, the number of solutions tested in parallel is of the order of the Avogadro number (that is 10^{23} molecules), opening interesting computational perspectives. Several authors have described how DNA computing could be used to solve difficult problems like boolean satisfiability [6]. Adleman himself has proposed an application to crack the DES encryption scheme [1]. DES is a standard symmetric encryption algorithm with a key of 56 bits, and a few test tubes of DNA would be enough to carry out a brute force attack on this cipher. Unfortunately, these approaches suffer from a number of drawbacks: (1) they are not always easy to implement in biochemistry, especially because they require purification steps; (2) they rely essentially on brute force because they do not easily make use of additional information that might be available about the problem. The price to pay for massive parallelism is a restricted flexibility in "programming" the DNA molecules. For cryptographic applications, this means that the "traditional" approach cannot take advantage of the known attacks on the weaknesses of a given algorithm.

Mao et al. [8] have recently shown that some degree of flexibility can be introduced in DNA computing while retaining the intrinsic advantage of massive parallelism. For that purpose they used DNA tiles that are a biochemical implementation of the mathematical concept of Wang tiles [15]. We will describe these objects in more detail in Section 2. Suffice it to say for now that the algorithmic self-assembly of Wang tiles is Turing Universal and that Mao et al. demonstrated the experimental feasibility of this concept. We feel that it is therefore appropriate to investigate in more detail to what extent the algorithmic self-assembly of DNA tiles can be used to solve problems that could not be practically solved using the "traditional" ap-

proach based on the self-assembly of linear DNA. In what follows, we demonstrate in particular that binary multiplication and cyclic convolution product are relatively straightforward to implement (Section 3). Furthermore, we show that the practical implementation of our ideas requires the creation of a finite number of tiles well within the reach of current combinatorial chemistry. Finally we discuss how our ideas could be used to implement a cryptanalytic attack on the well-known public-key crypto system NTRU (Section 4).

2 Algorithmic Self-Assembly

2.1 Wang Tiles

The concept of algorithmic self-assembly is closely related to that of Wang tiles. Wang showed that square tiles with colored edges can emulate a Turing machine, if they are allowed to assemble in a way that would cover the plane, according to additional rule that edges of the same color have to face each other [15]. This can be intuitively understood by thinking of a given row of tiles as representing a state of the Turing machine while the color encoding plays the role of the matching rules. This shows that computing using Wang tiles is universal [16, 13].

2.2 Physical Implementation of Wang Tiles

Recent advances in the field of materials science have enabled the experimental study of algorithmic self-assembly (abbreviated as ASA in the following). The first system, studied by Rothemund [12], was made of tiles whose edges were coated with materials of different hydrophobicity. The error rate was found to be unacceptably high, even though some expected distinctive features were observed. We remark that the hydrophobic/hydrophilic interaction used in these experiments was probably not specific enough to enforce proper edge matching (despite some very clever "ad-hoc" tricks used by the author). More recently Mao et al. [8] have shown that nanoscopic tiles can be manufactured using DNA that are the molecular equivalent of Wang tiles. These "Triple Crossover" tiles are made of several strands of DNA interwoven to create a square body made of DNA double helixes with single (reactive) strands of DNA sticking out from each edge of the tile. In the following we will refer to those single strands of DNA as sticky ends because they have the ability to bind to their Watson-Crick complement. This mechanism corresponds to the color matching rule in the abstract Wang tiles system. The experimental investigation focused on the XOR operation of two binary strings. The size of the problem was still relatively small, but the result turned out to be promising, with an error rate that was less than 2%. Therefore DNA seems to be the material of choice to implement ASA on a wider scale. Indeed, materials scientists have achieved a high degree of control over the nanostructures that can be built using DNA, and the interaction between single strands of DNA seems to be specific enough to enable a self-assembly with an acceptable error rate.

2.3 Our Perspective: Algorithmic Self-Assembly for Practical Problems

Even though ASA is universal, it does by no means follow that any problem can be practically addressed by this approach. Indeed, traditional DNA computing is also universal but, as mentioned above, the quantity of materials needed to perform a calculation prevents it to be used for anything but toy problems. Why is there any reason to believe ASA is a more interesting approach? The answer lies in the fact that DNA tiles can be more easily "programmed" to incorporate the constraints of a given problem. It is therefore possible to exercise some degree of control over the biochemical reaction occurring in the test tubes, thus avoiding the considerable waste of materials that characterize the traditional approach. Given the recent experimental developments mentioned above, we believe it is timely to reflect on the best use that could be made of ASA for practical purposes. Our approach is resolutely constructive: we try to provide examples where ASA turns out to be a practical way of solving otherwise difficult problems. This means that we have to depart from the only geometry that has been studied so far (square with four sticky ends). We give examples where a bigger number of sticky ends or a self-assembly not constrained to proceed in a plane turn out to be advantageous. To use a very bold analogy, this is reminiscent of the common situation in traditional computer science where a problem is straightforward to program in a given language (say C) while it is hard to address in another one (say assembly) [1].

3 Mathematical Operations in DNA

In this section we describe how to perform mathematical operations in DNA for two examples. First we show how to execute a multiplication in 2D. Then we introduce a method to carry out ASA in three dimensions to execute a cyclic convolution product. We give an abstract overview of each operation, and then go more into details. Note that we did not do any practical experiments.

[1] The analogy breaks down pretty quickly as one tries to give it a more formal shape, but we hope it is still useful to carry our message.

3.1 Multiplication

We implement the schoolbook method as shown in Figure 1. As example we use a multiplication of two 3-bit numbers. The binary input is given as vectors a and b with result r as sum of the corresponding rows under respect of carry overs. The spatial layout of the DNA after self-assembly is very similar to that of electronic circuits carrying out the same function [9].

$$
\begin{array}{cccccc}
 & \multicolumn{5}{c}{(a_2\,a_1\,a_0) \times (b_2\,b_1\,b_0)} \\
\hline
 & & & a_2\,b_0 & a_1\,b_0 & a_0\,b_0 \\
+ & & a_2\,b_1 & a_1\,b_1 & a_0\,b_1 & \\
+ & a_2\,b_2 & a_1\,b_2 & a_0\,b_2 & & \\
\hline
= & r_4 & r_3 & r_2 & r_1 & r_0
\end{array}
$$

Figure 1: Multiplication schoolbook method.

Figure 2 depicts the basic DNA tiles that are needed. Tile (1) is used for the actual execution. The original binary operand values are represented by a and b while s and c represent the intermediate sum and carry over, respectively. The result of this elementary operation, intermediate value and carry over, are denoted by s' and c'. It follows that

$$c' = (ab + c + s)/2,\ \ s' = (ab + c + s) \bmod 2$$

where integer division is used. There are 16 different input values determining the number of different tiles of this kind. Tiles (2) and (5) are used to represent operand bits. The connection to the next and previous input tile is denoted by j, while the final result of a column is connected at r. Tile (3) represents a result bit which will connect to the sticky end r of input tile (5) and the sum s' of tile (1). Furthermore we use frame tiles to limit the physical expansion of the execution. Frame tile (4) forwards the carry over value c to the next left column. Further auxiliary tiles are used (start and end).

Figure 3 shows the arrangement of the DNA tiles to perform a multiplication. Note that we pad the second operand b with 0-bits to make reading of the result easier, and that the result tile connected to b_0 is not part of the result. The body tiles are denoted by $v_{i,j}$, input tiles as a_i and b_j respective, and frame tiles by F. Extra tiles are needed as starting and end point, denoted as S and E. We understand that different kind of tiles need different sticky ends to avoid ambiguity. However, there are enough combinations available [4]. It is clear that this method can be applied to bigger operands, and that it does not require the operands to have the same length.

So far we have assumed that linear assemblies of input

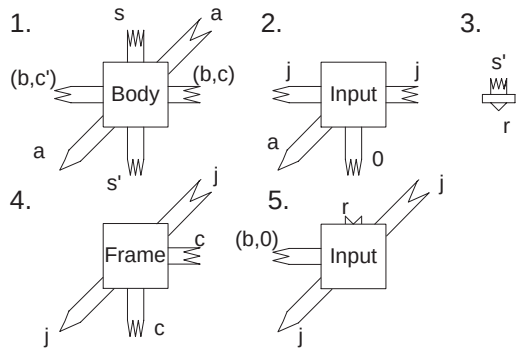

Figure 2: DNA tiles for multiplication.

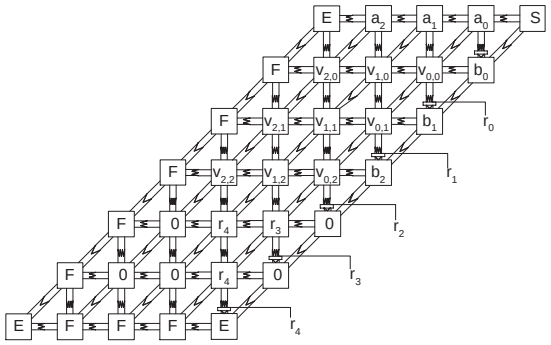

Figure 3: Multiplication in DNA.

tiles could be readily obtained. We now outline the way these inputs are "synthesized". Given a binary string of N bits, we need $2N$ different tiles indexed by their value (0 or 1) and their position within the string (the DNA sequence connecting one digit to the next one is of course unique for each pair of value and position). Creating a given input simply consists in picking out N such tiles with different indices. Note that, given the appropriate supply, if all the $2N$ tiles are mixed together it is possible to obtain the 2^N possible binary strings in a combinatorial fashion. By using non-identical concentrations for the two possible values at a given position, it is also possible to induce a probability distribution on the input strings. All in all, prior to any calculation involving two strings of length m and n, we need to synthesize sets of $m+n$ different input tiles[2], 16 body tiles, $4+2$ frame tiles, 2 result tiles, 3 end tiles, and 1 starting tile. Once the input strings have been synthesized our scheme requires only one reaction step. All the basic types of tiles are mixed together and the self-assembly can proceed. Reading the final result could be done using the reporter strand technique described in [8]. We note that in our case

[2]Note that combinatorial a and b requires $B(m+n)$ tile classes where B is the base of a and b, i.e., $2(m+n)$ for binary representation.

the reporter strand would have to run through the entire 2D lattice. Alternatively, one could imagine that each result tile would have a sticky end running perpendicular to the plane of self-assembly. This would allow the formation of a linear self-assembled structure above this plane, that could be used to produce a reporter strand whose size would scale linearly with the size of the solution[3]. Thus, even in the worst case, our multiplication scheme requires only two reaction steps and the number of different tiles required is growing linearly with the size of the problem.

3.2 Cyclic Convolution Product

After showing an example of ASA using relatively complex tiles to produce a straightforward 2D self-assembly, we now introduce an operation which can be performed more conveniently in 3 dimensions. First we define the cyclic convolution product. Let $F = \sum_{i=0}^{N-1} F_i x^i = [F_0, \ldots, F_{N-1}]$ be a polynomial or a vector of length N. Then the cyclic convolution product $\star$ of two vectors of length N is defined as [5]:

$$\begin{aligned} A \star B &= C \text{ with} \\ C_k &= \sum_{i=0}^{k} A_i B_{k-i} + \sum_{i=k+1}^{N-1} A_i B_{N+k-i} \\ &= \sum_{i+j \equiv k \bmod N} A_i B_j \end{aligned}$$

The $\star$ multiplication modulo q means that the coefficients C_k are reduced by q. From now on we will focus on the modulo product. Figure 4 gives a geometrical description of the convolution product. The input operands are the vectors a and b. The x and y axis describe the index of the operand bits. The figure shows the index k of c_k to which $a_i b_j$ contributes. By repeating the input vector a the result coefficient c_k can easily be obtained by adding the diagonal elements.

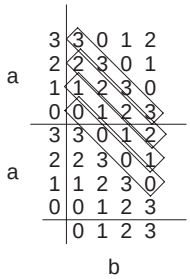

Figure 4: Geometrical description of convolution product.

[3]Note that the first "dummy" result tile comes in handy as a PCR primer.

We execute the convolution product in DNA according to the geometry just outlined. First we assemble the elementary multiplications in a "ground layer", then we grow the crystal to the third dimension to obtain the result. Figure 5 describes the body tile of the ground layer. It has two input ends a and b, forwards the input to the opposite side, and outputs the value ab using a sticky end pointing in the direction perpendicular to the plane of self-assembly [4]. Figure 6 shows how the ground layer is built. Again we use input tiles, frame tiles, and start and end tiles. Note that the first operand a is fixed since it has to be repeated.

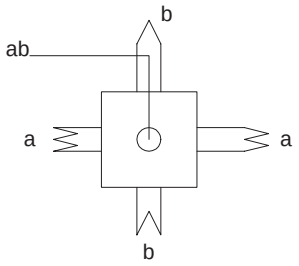

Figure 5: Body tile for convolution product.

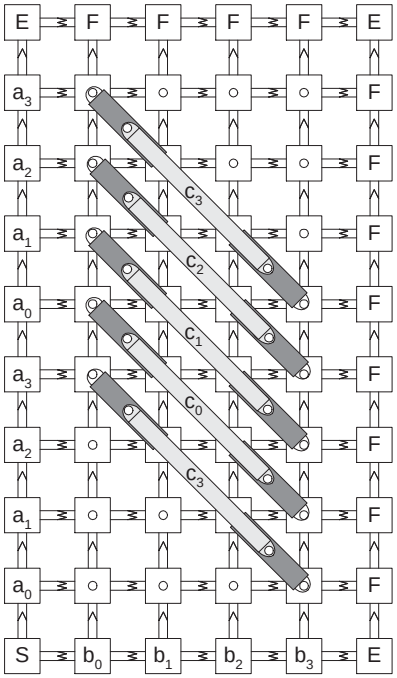

Figure 6: Cyclic convolution product in DNA.

To add the coefficients, we use bridges which are assembled for each layer beforehand. The implementation of the bridges ensures that the result coefficients

[4]It is depicted as a circle in Figure 5.

are modulo reduced. The bridges are built using connectors to the lower layer, a connector to the next layer, and spacer tiles[5]. Bridges broken down into their constitutive tiles are shown in Figure 7.

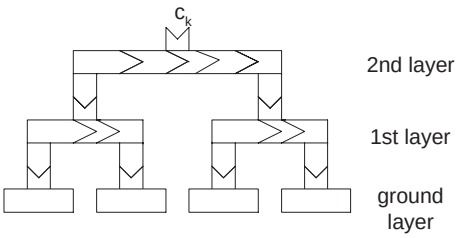

Figure 7: Bridge to add the coefficients.

The three dimensional arrangement of the bridges is shown in Figure 8. The bridges are arranged on top of the ground layer as outlined in Figure 6. The dark grey connections represent bridge connections in the first layer, while the light grey connections stand for second layer bridges.

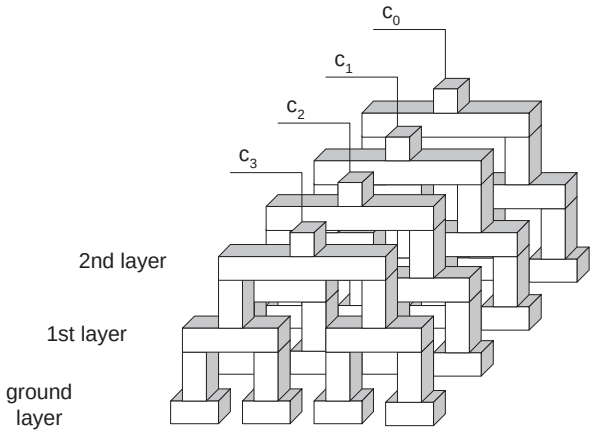

Figure 8: Bridges in the 3D space (connections in the ground layer are not pictured for simplification).

To simplify the bridge building operation we assume that the operands have a length which is a power of 2. To force the bridges to operate along the appropriate diagonal, it would be necessary to use a 2D lattice with a lower symmetry than the square symmetry used for convenience in Figure 6 [6]. The result of the operation can now be read at the sticky ends of the uppermost layer. The coefficient C_{N-1} appears twice and has to be ignored once. The diagonals which we do not consider will not assemble up to the highest layer.

Note that input coefficients are integers instead of binaries. Therefore the number of tiles needed is much larger than for the multiplication. Let $a_i \in \{0, \ldots, s-1\}$ and $b_j \in \{0, \ldots, t-1\}$. Synthesis of the input tiles requires $2N + N$ tile classes. Fixed a and combinatorial b requires $2N + tN$ tile classes though. Including start, end, frame, input, and body tiles we need $1+3+4+2N+N+st = 8+3N+st$ tile classes for the first layer. Remember that the bridge tiles perform addition modulo q. Therefore we need q^2 different bridge combinations, i.e., q^2 connector tiles to the upper layer and $2q$ connectors to the lower layer. Furthermore we need spacer tiles to build the bridges. Assuming that q is considerably larger than s and t the number of different tiles is in the order of q^2 even for combinatorial b. If $N = 2^x$ then our structure will consist of $x+1$ layers including the ground layer. Each layer will be grown one at at time, using bridges with the appropriate spacing. The final result will therefore be obtained in $x + 1$ steps.

3.3 More Practical Considerations

The first problem that needs to be addressed is that of the error rate during a computation. Theoretical considerations taking into account the thermodynamics of the system are clearly outside of the scope of this paper, and we will therefore only lead a qualitative discussion. The process of self-assembly within a plane, that we use for our multiplication scheme and as the first step in our cyclic convolution product, is in essence very similar to the construction of Mao et al. for their XOR product. It is therefore likely that an experimental error rate below 2% could be expected. Much higher error rates could be expected for the building of the successive layers in our convolution product because cooperativity is much lower in this direction (the number of neighbors is much lower which means less constraints). This would probably require the interaction energies between the sticky ends in this direction to be relatively high, in order to give a maximum energetic penalty to possible "orphan" sticky ends.

Any reader familiar with materials science will probably already have more than a few objections to our claims. Indeed, we must acknowledge that, to date, no DNA tile has been synthesized that could be used to implement our schemes directly. To what extent this will be true in the future is of course absolutely impossible to tell. We will simply refer the reader to the recent work accomplished by the group of Seeman [11] on the creation of DNA-based nanostructures and

[5] whose number depends on the layer under consideration

[6] To prevent that bridges are attached to the wrong tiles such that at some sticky ends there is no bridge attached at all one could use input tiles for a with alternate length in between.

let him decide for himself how far experimental science is from being able to implement our ideas. We do not believe that our computation schemes alone would be enough to motivate the considerable experimental work required to investigate the 3D ASA of DNA. But we should note that this technique is also very promising for the much more researched problem of protein crystallization. It is therefore not completely utopic to expect experimental progress on that front. Also, even though the total number of different types of tiles to be synthesized is not overwhelming and certainly within the reach of combinatorial chemistry techniques, even for operations on binary numbers of a few hundred bits, it remains to be seen which incentive an experimentalist could have to perform such an experiment. That's why we devote the next section of this article to show that it might be possible to implement an attack on a strong public-key crypto system using our strategy for the cyclic convolution product.

4 Application to Cryptography

In the mid 90's it was shown that DNA computing can be applied to break DES [3, 1]. These methods are based on a brute force attack. Using the parallel nature of DNA computing all possible keys are tested. For symmetric encryption ciphers like DES a brute force attack often is the only practical attack due to limited knowledge. However, for public-key methods brute force attacks are usually far out of computing power range because the key length is chosen according to the best known attack. which requires much less effort than brute force. DNA attacks are limited by the complexity of an attack step and the amount of DNA. In the following we will present the public-key system NTRU and a simple brute force attack in DNA. Then we present the execution of an attack on NTRU which reduces the amount of DNA by the square root. DNA tiles provide the appropriate flexibility to implement both attacks.

4.1 Overview of NTRU

4.1.1 Notation

In this section we will give a brief overview of NTRU. For further details see [5]. The NTRU system is based on a ring $R = \mathbb{Z}/(X^N - 1)$, three integers (N, p, q) and four sets $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_\phi, \mathcal{L}_m$ of polynomials of degree $N-1$ with integer coefficients. We assume that $gcd(p, q) = 1$, and that q is considerably larger than p. Elements $F \in R$ are written as a polynomial or vector

$$F = \sum_{i=0}^{N-1} F_i x^i = [F_0, F_1, \ldots, F_{N-1}]$$

Multiplication in R is done using the cyclic convolution product $\star$ as defined in Section 3.2. Multiplication modulo q means that the coefficients of the convolution product are reduced modulo q.

4.1.2 Key Creation

Assume two entities called Bob and Alice who want to exchange messages over an insecure channel. First Bob chooses elements $f \in \mathcal{L}_f$ and $g \in \mathcal{L}_g$. For simplicity we assume that f has coefficients in $\{0, 1\}$ and that g has coefficients in $\{0, \ldots, s-1\}$. The polynomial f is chosen such that it has exactly d coefficients of value 1 and $N-d$ coefficients of value 0. Bob computes $f_q^{-1} \equiv f^{-1} \bmod q$ and $h \equiv f_q^{-1} \star g \bmod q$. Bob's private key is the polynomial f and his public key is h.

4.1.3 Encryption

To encrypt a plain text message $m \in \mathcal{L}_m$ using Bob's public key h, Alice selects a random element $r \in \mathcal{L}_\phi$ and computes the cipher text $e \equiv (r \star h + m) \bmod q$.

4.1.4 Decryption

To decrypt the cipher text e using the private key f, Bob first computes $a \equiv f \star e \bmod q$ where he chooses the coefficients of a in the interval from $-q/2$ to $q/2$. Now Bob recovers the plain text message as $m \equiv (f^{-1} \bmod p) \star a \bmod q$.

4.2 Brute Force Attack

The goal of the attack is given all parameters (N, p, q), the sets $\mathcal{L}_f, \mathcal{L}_g, \mathcal{L}_\phi, \mathcal{L}_m$, and a public key h to recover the private key f. Let us assume that polynomials in $\mathcal{L}_g$ have coefficients in $\{0, \ldots, s-1\}$. As before any $f \in \mathcal{L}_f$ has d coefficients of value 1 and $N-d$ coefficients of value 0. An attacker can recover the private key by trying all possible $f \in \mathcal{L}_f$ and testing if $g' = f \star h \bmod q$ has small entries, i.e., if the coefficients are between 0 and $s-1$. Similarly, an attacker can try all $g \in \mathcal{L}_g$ and test if $f' = g \star h^{-1} \bmod q$ has only coefficients 0 or 1. In practice, $\mathcal{L}_g$ is smaller than $\mathcal{L}_f$, so the security is determined by the number of elements in $\mathcal{L}_g$.

The attack can be implemented in DNA as follows. For all $g \in \mathcal{L}_g$ compute the cyclic convolution product $g \star h^{-1} \bmod q$ as explained in Section 3.2. Choose h^{-1} as the operand which is repeated. Use the massive parallelism of DNA to compute the convolution product of h^{-1} with all the possible g. Reading of the operation result is done by the reporter strand method. Among all the results, the one consisting only of 0 or 1 [7] is the

[7] Remember that each digit of the result can take q

private key. To get it we need to run $q-2$ separation steps.

This attack does not scale up well. It is limited by q which determines the number of different tiles. A typical value for q is 64 or 128 [5] which means that more than 4,000 different tile classes are needed. Another restriction is given by the total amount of DNA. The number of DNA tiles to be used in the computation cannot be expected to be much more than the Avogadro number (about 10^{23}). Therefore this kind of attack is roughly limited to 2^{80} different possibilities for g. Since the coefficients of g are not limited to binary values there can be 2^{80} different possible polynomials of length 64. Usually the key space is defined as the set of possible keys. In our case we extend the definition such that the key space is the set which is used for an attack, i.e., for NTRU this is usually $\mathcal{L}_g$ since it is smaller than $\mathcal{L}_f$. The key security is defined as the number of steps, or in our case different inputs, that have to be performed or tried before the key is found using the best known attack. The best known attack is shown in the next section and reduces the effort by a square factor, i.e., the key security is the square root of the number of elements in the key space. Thus we can break an NTRU system having a key security of 2^{40} for a proper value s. However, a typical key security for NTRU in high security scenarios is about 2^{80}, i.e., $\mathcal{L}_g$ has 2^{160} elements. In the next section we give a future perspective how this can be achieved.

4.3 Meet-in-the-Middle Attack

The meet-in-the-middle attack reduces the effort to find the private key. Compared to a brute force attack this attack reduces the amount of DNA which is required for a successful attack by the square root, or in other words the key space which can be broken is quadratic in size. We will give a brief overview of the attack. Detailed information is given in [14]. Remember that the private key f has exactly d ones and $N-d$ zeros. The idea of the attack is to search for f in the form (f_1, f_2) where f_1 and f_2 each have $d/2$ ones and are $N/2$ in length. Then try all possibilities for f_1 and f_2 such that $(f_1, f_2) \star h \bmod q$ has coefficients between 0 and $s-1$. This can be done efficiently as follows.

Choose at random $N/2$ of the N possible positions in f. Assume that $d/2$ of these $N/2$ positions have ones in the actual private key f. The probability that this assumption holds is approximately 1 to $\sqrt{d}$, so the following has to be repeated around $\sqrt{d}$ times before f is found. Now relabel the positions such that the chosen $N/2$ positions determine the vector f_1 and the other $N/2$ positions f_2. The next step is to enumerate over f_1. Usually this takes only $\binom{N/2}{d/2}$ steps but in DNA we probably have to iterate over all $2^{N/2}$ binary vectors. Since d is chosen such that the key space is very large the relative difference is very small. First $(f_1, 0) \star h \bmod q$ is computed and put into a bin based on its first k coefficients. If the convolution product has coefficients $F_0, \ldots, F_{k-1}$ then it is put into a bin $(I_{j_0}, \ldots, I_{j_{k-1}})$ where $I_{j_i} \supset F_i$ are integer intervals. The size of the intervals are determined by k. Then all possible values for f_2 are enumerated. The convolution product $-(0, f_2) \star h \bmod q$ is put into a bin $(J_{j_0}, \ldots, J_{j_{k-1}})$ that is defined by intervals that are slightly larger as the previous ones (each bin is exactly by s-1 larger). Finally the bins are compared. If the assumption about the position of the ones in f in the first step was right then there is a matching pair f_1 and f_2 such that f_1 is in $(I_{j_0}, \ldots, I_{j_{k-1}})$ and f_2 in $(J_{j_0}, \ldots, J_{j_{k-1}})$, and the private key can be derived as $f = (f_1, f_2)$.

In DNA the attack is executed as follows. First choose $N/2$ random positions in f. The marked positions are represented by f_1 which is assembled using DNA tiles such that the marked positions are chosen combinatorial and the unmarked positions are set to 0. This can easily be done by encoding the position into the DNA tiles that represent f_1 as described before. Execute $(f_1, 0) \star h \bmod q$ in parallel for all possible combinations of f_1 as explained in Section 3.2. The polynomial h is the fixed operand which is repeated. Now construct DNA tiles for f_2 in the same manner as before but set marked bits to 0 and iterate over unmarked bits, and execute $-(0, f_2) \star h \bmod q$. To put a product into a bin we use special DNA tiles with two sticky ends that translate an integer value into the corresponding interval. These tiles are different for the two convolution products in the sense that the interval sizes are different. Tiles for the first convolution product can connect to tiles for the second product with the same intervals, i.e., tiles representing I_{j_i} will be glued to tiles representing J_{j_i}. Apply these tiles to the convolution products. Assuming that the mobility of the DNA supra molecular assemblies is not too small, two of them will stick together. If such a tandem structure can be found the original assumption was right, and the private key can be determined by reading the input tiles f_1 and f_2. The actual reading stage is problematic here, as the reporter strand method likely does not seem to work. We suggest another approach where the DNA structures are first filtered according to their molecular weight, and those corresponding to tandem units are examined by atomic force microscopy[8].

values.

[8] If the recent developments in coupled NMR and AFM become mainstream, then reading could be done by coor-

Assuming that a brute force attack can be mounted to break a key security of 2^{40} the described meet-in-the-middle attack in DNA might break systems with a key security of 2^{80}. However, many assumptions are very optimistic for the near future. Furthermore we understand that using a higher security level, e.g., a key security of 2^{285} as proposed in [5] puts public-key systems like NTRU far out of range for a successful cryptanalysis in DNA.

5 Conclusions

We have presented two computation schemes for the binary multiplication and cyclic convolution product using the algorithmic self-assembly of DNA tiles. For that purpose, we introduced new conceptual designs for DNA tiles that should allow for a practical implementation of these operations. Indeed, we emphasize the fact that even though DNA tiles are by themselves universal, tiles with different designs will perform very differently on a given problem: designing effective DNA tiles for a given computation can be thought as "DNA programming". The most interesting feature of our system of DNA tiles is that it turns out to be flexible enough to go beyond a simple brute force algorithm: it would indeed be possible to use it to implement an attack on a public-key crypto system. Among the open questions, we have to acknowledge that we do not have any estimations on the expected error rate. We are currently trying to address this issue.

References

[1] L.M. Adleman, P.W.K. Rothemund, S. Roweis, E. Winfree. On applying molecular computation to the Data Encryption Standard. In *Proceedings of 2nd DIMACS workshop on DNA based computers*, held at Princeton University, USA, 1996.

[2] L.M. Adleman. Molecular Computation of Solutions to Combinatorial Problems. *Science*, vol. 266, pages 1021–1024, 1994.

[3] D. Boneh, C. Dunworth, and R.J. Lipton. Breaking DES using a molecular computer. *Tech. Report CS-TR-489-95*, Princeton University, USA, 1995.

[4] U. Feldkamp, W. Banzhaf, H. Rauhe. A DNA Sequence Compiler. In *Proceedings of 6th DIMACS Workshop on DNA Based Computers*, held at University of Leiden, Netherlands, 2000.

dinating atoms with very different resonance frequencies to the input tiles.

[5] J. Hoffstein, J. Pipher, J.H. Silverman. NTRU: A Ring-Based Public Key Cryptosystem. In *Proceedings of ANTS III*, vol. 1423 of LNCS, pages 267–288, Springer-Verlag, 1998.

[6] R.J. Lipton. DNA Solution of Hard Computational Problems. *Science*, vol. 268, pages 542–545, 1995.

[7] Q. Liu, L. Wang, A.G. Frutos, A.E. Condon, R.M. Corn, L.M. Smith. DNA Computing on Surfaces. *Nature*, vol. 403, pages 175–179, 2000.

[8] C. Mao, T.H. LaBean, J.H. Reif, and N.C. Seeman. Logical computation using algorithmic self-assembly of DNA triple-crossover molecules. *Nature*, vol. 407, pages 493–496, 2000.

[9] B. Parhami. *Computer Arithmetic: Algorithms and Hardware Designs*. Oxford University Press, New York, 2000.

[10] Q. Ouyang, P.D. Kaplan, L. Shumao, A. Libchaber. DNA Solution of the Maximal Clique Problem. *Science*, vol. 278, pages 446–449, 1997.

[11] N.C. Seeman et al. New Motifs in DNA Nanotechnology. In *Proceedings of 5th Foresight Conference*. To be published in Nanotechnology (2001).

[12] P.W.K. Rothemund. Using lateral capillary forces to compute by self-assembly. In *Proceedings of National Academy of Science*, vol. 97n3, pages 984–989, 2000.

[13] P.W.K. Rothemund, E. Winfree. The Program-Size Complexity of Self-Assembled Squares. In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, May 2000, Portland, USA.

[14] J.H. Silverman. A Meet-In-The-Middle Attack on an NTRU Private Key. *NTRU Technical Report #4*, 1997.

[15] H. Wang. Proving theorems by pattern recognition. II. *Bell System Technical Journal*, vol.40, pages 1–42, 1961.

[16] E. Winfree, F. Liu, L.A. Wenzler, N.C. Seeman. Design and Self-Assembly of two-dimensional DNA Crystals. *Nature*, vol. 394, pages 539–544, 1998.

A DNA-based three-state device

Bernard Yurke
Bell Laboratories
Lucent Technologies
Murray Hill, NJ 07974

Friedrich C. Simmel*
Bell Laboratories,
Lucent Technologies
Murray Hill, NJ 07974

Abstract

A simple DNA-based nanomechanical device is presented which can be switched between three different conformations by the addition of appropriate signal molecules. Two of these conformations are mechanically rigid. The device consists of two arms formed by double-stranded DNA, connected by a single-stranded hinge. The device can be in a closed or stretched configuration, and in an intermediate relaxed state. The operation of the device is monitored by fluorescence resonance energy transfer and gel electrophoresis experiments. The closing of the supramolecular arms is affected by the salt concentration, probably due to electrostatic interactions of the device with itself.

1 INTRODUCTION

Its molecular recognition properties make DNA a promising molecule for the development of molecular nanotechnology. DNA has already been utilized to build a number of complex nanoscale structures [1, 2, 3]. Moreover, in recent experiments the remarkable properties of DNA could even be used to induce motion on a molecular scale [4, 5, 6, 7, 8, 9, 10]. The basis for molecular engineering with DNA is the – under appropriate conditions – highly specific base pair (bp) recognition between the DNA bases adenine (A) and thymine (T), and between guanine (G) and cytosine (C) [11]. By the choice of base sequence, DNA strands can be made "sticky" or non-interacting. The "programmability" of these interactions renders DNA particularly interesting as a self-assembly molecule as it offers much more design freedom than available in other self-assembly approaches [12]. In the case of nanomechanical devices, the information-carrying character of DNA is not only utilized for the assembly of the devices but also as a means of addressing (and therefore controlling) their conformational transitions. Here we present a simple DNA-based device which can be switched between three different mechanical states. Starting from a relaxed intermediate conformation, the device can either be transferred to a stretched or to a tightened state, depending on the "fuel" or "signal" strand added to it. From these configurations the device can be returned to the original state, and therefore this simple molecular machine can be controllably cycled through its different states. Earlier DNA constructs like DNA tweezers [5], DNA actuator [6, 8] or DNA scissors [7] based upon the same operation principle were switchable only between a flexible and a rigid conformation. A combination of tweezers and actuator resulted in a three-state device (TSD) which could be switched between two rigid and one flexible state [10]. In the present work, we provide additional evidence for the proper operation of this TSD. We also investigate the influence of ionic strength on the different conformations of the device and thereby demonstrate the flexibility of the relaxed state as opposed to the rigidness of the stretched and the tightened state.

2 DESCRIPTION OF THE DEVICE

The three-state device in its relaxed conformation is assembled from the 40 nucleotide (nt) long DNA strand Q and the 84 nt long strand R (Fig. 1). Q and R form two 18 bp long "arms," connected by 4 nt and 48 nt long single-stranded sections. The ≈ 13.6 nm long arms are mechanically stiff elements on this length scale, as they are considerably shorter than the persistence length of double-stranded DNA (about 50 nm [13]). The single-stranded sections, however, are rather flexible [14]. During the device's operation the 4 nt section acts as a hinge, whereas the long 48 nt section is used to address and induce the motion of the

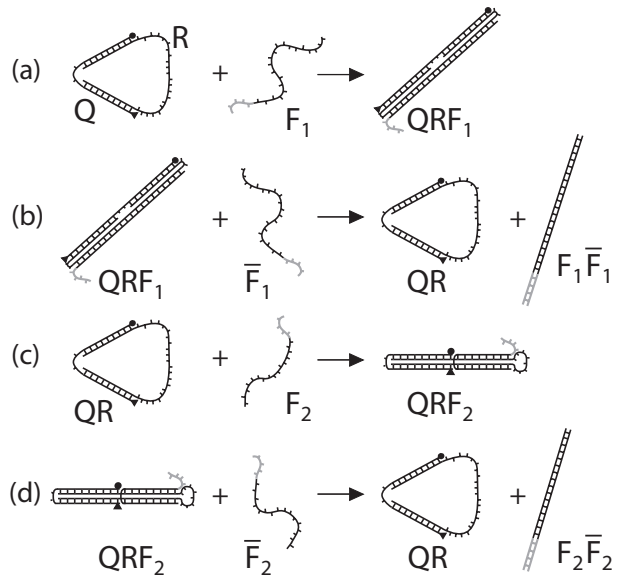

Figure 1: Schematic representation of the operation principle of the three-state device: (a) The TSD consists of two single strands of DNA, Q and R. They hybridize together to form two double-stranded arms connected by a hinge and a long single-stranded section. The TSD is stretched by the fuel strand F_1 which hybridizes to the single-stranded part of QR. (b) The removal strand $\bar{F}_1$ can attach to an unhybridized section of F_1 (depicted in gray) and remove F_1. This restores the relaxed conformation. (c) F_2 can hybridize with QR in a different manner and thus close the arms of the device. (d) Similar to (b) the relaxed state can be restored by the addition of $\bar{F}_2$.

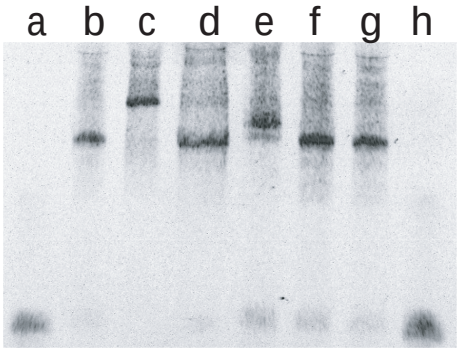

Figure 2: Fluorescence image of the result of a 10% polyacrylamide gel electrophoresis run of the TSD and its components. DNA strands run from the top to the bottom of the gel; smaller complexes run faster. Lanes (a) and (h) contain strand Q, lanes (b) and (g) the relaxed TSD (Q + R). Lane (c) contains the stretched TSD (QR + F_1). In lane (d) the fuel strand has been removed again (QRF$_1$ + $\bar{F}_1$). In lane (e) the TSD is in its closed conformation (QR + F_2). The band shift is smaller than in lane (c) as F_2 is shorter than F_1. Finally, in (f) the closing strand has been removed to restore the relaxed state of the TSD (QRF$_2$ + $\bar{F}_2$). Only complexes containing the fluorescently labeled strand Q are visible in this image. The smears above the main bands are caused by multimerization products. Several Q strands can be linked together by R strands and several QR complexes can be crosslinked by fuel strands. This effect has been extensively discussed in Refs. [5, 6, 8]

TSD.

The conformational transitions of the TSD are depicted in Fig. 1: The relaxed device (QR) can be transferred into the stretched configuration (QRF$_1$) with the help of the 48 nt long fuel strand F_1 (Fig. 1 (a)). As F_1 hybridizes with the long single-stranded section of QR, the formation of the DNA duplex straightens the TSD. In Fig. 1 (b) the TSD is returned to its relaxed state. F_1 is equipped with a short eight base long section (drawn in gray) which does not hybridize to QR. At this overhang section F_1 can be attacked by $\bar{F}_1$ which is a strand complementary to F_1. The fuel strand F_1 is removed from QRF$_1$ by $\bar{F}_1$ in a process known as "branch migration" [15]. In this process, both QR and $\bar{F}_1$ compete for binding to F_1. During branch migration, the branch point, at which the three strands R, F_1 and $\bar{F}_1$ meet, performs a random walk along R. When the branch point makes its first passage, the strand displacement process is completed and the "inert" waste product $F_1\bar{F}_1$ falls off the now relaxed TSD.

The tightening (or closing) motion of the TSD is shown in Fig. 1 (c). Here, a different 40 nt long fuel strand F_2 is used which hybridizes to R in a manner reverse to that of the straightening transition. F_2 pulls the two arms of the device close together. A short ring-like section of R is left unhybridized to facilitate the attack of the second removal strand $\bar{F}_2$ which again restores the relaxed configuration by branch migration (Fig. 1 (d)). Using the appropriate fuel and removal strands, the TSD can be switched arbitrarily between its three conformations QR, QRF$_1$, and QRF$_2$.

3 METHODS

Oligonucleotides were synthesized, labeled and purified by Integrated DNA Technologies, Inc. (IDT). The sequences for the strands can be found in references [6, 10]. The operation of the TSD is checked

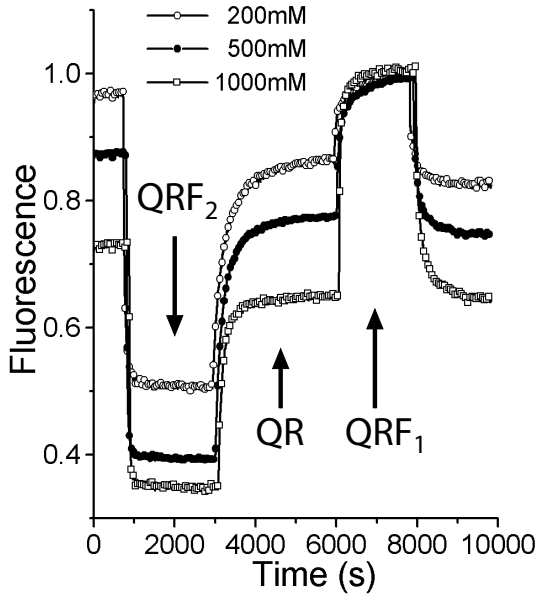

Figure 3: The TSD operation is monitored with fluorescence measurements at three different salt concentrations: The devices start in the relaxed configuration (QR). By the addition of fuel strand F_2 they are transferred into the closed conformation (QRF_2). From there, they are returned to state QR with the removal strand $\bar{F}_2$. Subsequently, the TSD is brought into the stretched state QRF_1 by the addition of fuel strand F_1. Removal of this strand with its complement $\bar{F}_1$ completes one operation cycle. The changes in the fluorescence levels are due to FRET and correspond to changes in the separation between the fluorescent dyes attached to the arm of the TSD (see Fig. 1).

with gel electrophoresis and fluorescence resonance energy transfer experiments. For the latter, strand Q is labeled at the 5' end with the fluorescent dye TET (2',4',5',7'-Tetrachloro-5(6)-carboxyfluorescein) and at the 3' end with TAMRA (N,N,N',N'-Tetramethyl-5(6)-carboxyrhodamine) (symbolized by the black circles and triangles in Fig. 1). TET and TAMRA form a fluorescence resonance energy transfer (FRET) pair [16] with a Förster distance R_0 of approximately 5 nm. The emission band of TET (the "donor") and the absorption band of TAMRA (the "acceptor") overlap and TET can transfer its excitation energy nonradiatively to TAMRA. This quenches the TET fluorescence. The energy transfer efficiency depends strongly on the distance between donor and acceptor. At a distance R_0 between the dyes, the efficiency of energy transfer is 50%. To estimate dye separations, we use an experimentally obtained calibration curve [5, 17]. For the FRET experiments, the relaxed devices are assembled by mixing stoichiometric amounts of 25 μM solutions of strands Q and R in TE buffer (10 mM Tris(hydroxymethyl)-aminomethane (Tris), pH 8.0, 1 mM ethylene diamine tetraacetic acid (EDTA)) and diluting these mixtures in reaction buffer to a final concentration of 1 μM. As reaction buffers we used TE buffer with added salt (TE/200 mM NaCl, TE/500 mM NaCl, TE/1M NaCl) or "physiological buffer" (potassium phosphate buffer, pH 7.2, with $[K^+]$=140 mM, $[Na^+]$=10 mM and $[Mg^{2+}]$=0.5 mM (intracellular values) [18]). The TSDs are switched between their different conformations by the addition of stoichiometric amounts of 25 μM fuel or removal strands, followed by rapid mixing. The fluorescence of TET is excited with light from an Argon ion laser (λ = 514.5 nm) chopped at a frequency of 130 Hz. The fluorescence light is filtered with a 10 nm bandpass filter centered at 540 nm and detected with a Si photodiode at the chopper frequency.

For the gel electrophoresis experiments, strands are mixed in stoichiometric amounts at a concentration of 2.5μM in TE buffer. Polyacrylamide gels were cast at a concentration of 10% and run at 10 V/cm and T = 20°C. To visualize the bands, the gels were illuminated using the laser system described above and photographed through a bandpass filter with a digital camera.

4 RESULTS

In Fig. 2 the result of a gel run in a 10% polyacrylamide gel is displayed. The band shifts correspond to the molecular weight changes of the TSD during its operation. Lanes (a) and (h) contain only strand Q, whereas lanes (b) and (g) contain the relaxed conformation QR. In the remaining lanes the TSD is cycled once through its states: Straightened (lane c), relaxed (lane d), closed (lane e) and relaxed again (lane f). The band shifts visible on the gel image correspond to the increase or decrease of the molecular weight of the TSD with the attachment or removal of fuel strands.

Typical FRET signals collected during the operation of the TSD are shown in Fig. 3 for three different salt concentrations. The fluorescence of the device in its straightened conformation (QRF_1) has been normalized to one, as this conformation is least influenced by the salt concentration. In the straightened conformation, the sample displays its maximum fluorescence, as in this state the two dyes are furthest apart from each

other and therefore FRET is least efficient. In the intermediate state (QR), TET and TAMRA are closer to each other, and therefore the fluorescence is partly quenched. This effect is stronger for higher salt concentrations. In the closed state QRF_2, the two dyes are brought very close to each other, and therefore the TET fluorescence is quenched most efficiently. In this conformation, the fluorescence of the device is also lower for higher salt concentrations. However, its sensitivity is not as strong as for the relaxed state. The transitions between the conformations display second order kinetics with reaction half-times on the order of $t_{1/2} = 10$ s – 100 s. The reactions are generally faster for higher salt concentrations [19].

For possible biomedical applications it is important to investigate whether nanomechanical devices as the TSD are operable under physiological conditions. To this end, a wide range of salt concentrations, pH values and temperatures have been investigated. It is found that the temperature dependence of the reaction kinetics exhibits the expected Arrhenius-like behavior. The fluorescence signal changes are highest for high (monovalent) ion concentrations and low temperatures, whereas they are relatively insensitive to changes in the pH value [19]. For Fig. 4, the TSD has been operated in physiological buffer at $T = 37$°C. The half-times for completion of the reactions are comparable to the ones for the data presented above: The lower salt concentrations slow the reactions down, whereas the higher temperature speeds them up. The changes in the fluorescence intensity are considerably less under physiological conditions than at lower temperatures and higher salt concentrations.

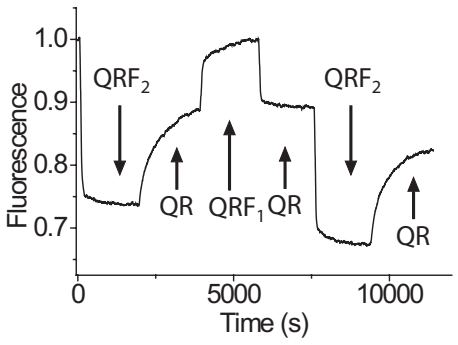

Figure 4: The TSD is also operable under physiological conditions (physiological buffer at $T = 37$°C). Under these conditions the reaction kinetics and the fluorescence signal levels are altered. Higher temperature speeds up the reactions; the lower salt concentrations have the opposite effect. Also, the fluorescence intensity changes decrease with higher temperature.

5 DISCUSSION

The gel electrophoresis band shifts in Fig. 2 and the fluorescence changes in Fig. 3 are in full agreement with the proposed operation scheme for the TSD from Fig. 1. From the fluorescence levels one can deduce the distances between the dyes for the different conformations at different salt concentrations. In the stretched state their separation is 13.6 nm which is just the length of a 40 bp double helix. In the relaxed state the separations between the dyes are 8.2 nm, 6.2 nm and 5.1 nm for sodium ion concentrations of 200 mM, 500 mM and 1 M, respectively. In the closed state the corresponding separations are approximately 3.7 nm, 2.7 nm and 2.0 nm. This shows that at high salt concentrations the device can be switched between two rigid conformations which is a major improvement over earlier devices based on the same operation principle [5, 6]. These devices could only be switched between one rigid and one flexible conformation. The

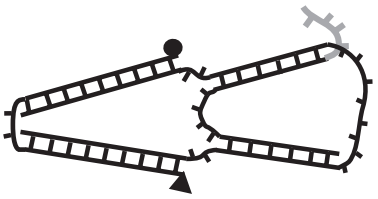

Figure 5: Model of closed TSD at low salt concentration. Reduced screening leads to mutual electrostatic repulsion of the arms. This partly unzips the fuel strand F_2 from R.

flexibility of the relaxed configuration would severely constrain the performance of the two-state devices if operated against an externally applied force. With the three-state device and its two rigid conformations this limitation is overcome. It can be expected that the TSD is capable of actually performing work against an external force if it is switched between the tightened and stretched configurations. However, our results indicate that at low salt concentrations the closed state of the TSD also becomes distorted. Due to the reduced screening of electrostatic interactions at low ion concentrations, the arms of the device begin to push on each other and the distance between the arms increases. This explains the increase of the fluorescence intensity in the closed state for low salt concentrations. At low enough salt concentrations, the mutual repulsion becomes strong enough to break base pairs (Fig. 5). A quantitative treatment of this effect might provide a possibility to calibrate the forces generated by the TSD. In the relaxed state, the reduced screening at lower salt concentrations probably results in the

stiffening of the single-stranded section of strand R which also pushes the dyes further apart.

The TSD is an example of a chemically addressable molecular device. The fuel strands F_1 and F_2 can be thought of as signal strands with which one can externally control the conformational changes of the device. Based on this principle, more elaborate molecular devices could be developed which are capable of a larger number of conformational changes controllable in a similar fashion as the TSD. Alternatively, a population of devices could be devised, which are capable of interacting with each other with the help of signal strands. A possible application of this might lie in the assembly of nanoscale components in a certain spatiotemporal order, e.g., the installation of molecular electronic components. Interacting DNA devices might also be used to construct chemical reaction networks performing particular tasks. It has to be mentioned that the construction of complex machines or networks faces similar problems as DNA-based computation: The number of machines (or conformations, interactions, etc.) is limited by the number of distinct base sequences available which robustly assemble into the correct structures without unwanted cross-hybridizations. Otherwise, incorrectly assembled devices, extensive crosslinking and high error rates in addressing the devices will make an operation of the machines impracticable.

Apart from nanotechnology and molecular electronics, possible applications of DNA nanodevices can also be found in the biomedical area. Molecular devices could be used as drug delivery systems or biosensors. With results obtained in the field of DNA-based computing, one could even combine the mechanical operation of the devices with information processing activities. The stable operation of our DNA devices under physiological conditions makes such a biomedical application quite plausible, even though for such an application the usual problems of gene delivery would have to be solved first.

6 CONCLUSIONS

We have demonstrated the construction and operation of a DNA-based molecular device which is switchable between three different conformations. Two of these conformations are mechanically robust. Due to the highly charged nature of DNA, the device operation is strongly affected by changes in the salt concentration of the reaction buffer. However, the device is still operable under physiological conditions.

Acknowledgments

Financial support by the Alexander von Humboldt Foundation through the Feodor Lynen program is gratefully acknowledged (F.C.S).

*present address: Center for NanoScience and Sektion Physik, Ludwig-Maximilians-Universität, Geschwister-Scholl-Platz 1, 80539 München, Germany.

References

[1] J. Chen and N. C. Seeman, Nature **350**, 631 (1991).

[2] C. Mao, W. Sun, and N. C. Seeman, Nature **386**, 137 (1997).

[3] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, Nature **394**, 539 (1998).

[4] C. Mao, W. Sun, Z. Shen, and N. C. Seeman, Nature **397**, 144 (1999).

[5] B. Yurke, A. J. Turberfield, A. P. Mills Jr., F. C. Simmel, and J. L. Neumann, Nature **406**, 605 (2000).

[6] F. C. Simmel and B. Yurke, Phys. Rev. E **63**, 041913 (2001).

[7] J. C. Mitchell and B. Yurke, DNA 7, Seventh International Meeting on DNA-Based Computers, Springer Lecture Notes in Computer Science, in print (2002).

[8] F. C. Simmel and B. Yurke, DNA 7, Seventh International Meeting on DNA-Based Computers, Springer Lecture Notes in Computer Science, in print (2002).

[9] H. Yan, X. Zhang, Z. Shen, and N. C. Seeman, Nature **415**, 62 (2002).

[10] F. C. Simmel and B. Yurke, Applied Physics Letters **80**, 883 (2002).

[11] V. A. Bloomfield, D. M. Crothers, and I. Tinoco Jr., *Nucleic Acids*, University Science Books, Sausalito (2000).

[12] J. J. Storhoff and C. A. Mirkin, Chem. Rev. **99**, 1849-1862 (1999).

[13] S. B. Smith, Y. Cui, and C. Bustamante, Science **271**, 795 (1996).

[14] B. Tinland, A. Pluen, J. Sturm, and G. Weill, Macromolecules **30**, 5763 (1997).

[15] I. G. Panyutin and P. Hsieh, Proc. Natl. Acad. Sci. USA **91**, 2021 (1994).

[16] L. Stryer and R. P. Haugland, Proc. Nat. Acad. Sci. USA **58**, 719 (1967).

[17] F. C. Simmel and B. Yurke, Proc. SPIE **4332**, 419 (2001).

[18] B. Alberts, D. Bray, J. Lewis, M. Raff, K. Roberts and J. D. Watson, *Molecular Biology of the Cell*, 3rd ed., Garland Publishing, New York (1994).

[19] F. C. Simmel, B. Yurke, and R. J. Sanyal, submitted to J. Nanosc. Nanotech. (2002).