Batching DNA Computations

Dan Boneh
dabo@cs.princeton.edu

Richard J. Lipton*
rjl@cs.princeton.edu

Department of Computer Science
Princeton University
Princeton, NJ 08544

Abstract

In this short note we point out that in some cases batching computations on a molecular computer is very cheap. This is especially useful for breaking cryptosystems and can be useful for other applications as well.

Recently Lipton [1] showed that the formula satisfaction problem can be solved by a DNA computer. This result holds for formulas with at most 70 inputs. Lipton's result was then used in [2] to show that the DES cryptosystem can be broken in roughly 4 month of work in the lab. It was pointed out that under chosen plain-text attack it only takes one day to break DES with some preprocessing. In this short note we expand on this point.

Let $f : \{0,1\}^{70} \to \{0,1\}^{70}$ be some one-to-one function on 70 inputs. The fact that f is one-to-one implies that it has an inverse which we denote by f^{-1}. Suppose f can be computed using a circuit of size s. The results of [3] show that using $O(s)$ biological operations it is possible to build a soup of strands representing all elements in the set

$$\{(x, f(x)) \mid x \in \{0,1\}^{70}\}$$

We refer to this solution as T_f. The point is that given T_f and a value $f(x)$ one can easily recover x. This can be done by extracting all strands in T_f containing the string $f(x)$ on its right side. The extracted strands will represent the pair $(x, f(x))$. The initial section of the extracted strands can be sequences and hence x can be found. This simple process takes only several hours.

The simple procedure above shows that given T_f one can compute $f^{-1}(x)$ very quickly. In this note we point out that if one had to compute $f^{-1}(x_0), \ldots, f^{-1}(x_m)$ then one can perform the computation in time

$$O(s) + O(m) \qquad (1)$$

$O(s)$ is the time it takes to construct the solution T_f and $O(m)$ is the time to perform the m table lookups for the required values. Note that once the solution T_f has been constructed, evaluating $f^{-1}(x)$ takes constant time.

We refer to the process of evaluating the function f at many points at once as *batching*. On a conventional computer, if it takes time s to evaluate a function, then the time it takes to evaluate the

*Supported in part by NSF CCR–9304718.

function at m points is $O(ms)$. This should be contrasted with (1). Thus batching is *much* easier on a DNA computer than on a conventional computer. This is the main point of this note.

The implication of (1) to cryptography is extremely important. Let $g(M, k)$ be a private key cryptosystem; the encryption of the message M using the key k is $g(M, k)$. Suppose an eavesdropper wishes to tap a conversation protected using the cryptosystem g. Under plain-text attack the eavesdropper can generate a pair (M_0, E_0) where he gets to choose M_0. Define the function $f(k)$ to be $f(k) = g(M_0, k)$. Then to find k the eavesdropper needs to compute $f^{-1}(E_0)$.

When the key k is at most 70 bits long the batching argument above shows that some large company can spend the time to construct the solution T_f. The company can then sell the solution to anyone who wishes to quickly break the cryptosystem g. This argument shows that due to the existence of DNA computers, cryptosystems had better use keys which are much longer than 70 bits. Some cryptosystems which are currently used do not use large enough keys. Examples are DES and clipper.

To conclude, we note that batching the evaluation of one way functions is cheap when using a DNA computer. This phenomenon is an example where a DNA computer has a clear advantage over a conventional machine. Other applications where batching is useful will provide more examples to which DNA computers are especially well suited.

References

[1] R. Lipton, "Using DNA to solve NP-Complete Problems", Science 268:542-545 (Apr. 28) 1995.

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

[3] D. Boneh, C. Dunworth, R. Lipton, J. Sgall, "On the Computational Power of DNA", submitted to Discrete and Applied Math.