ON THE NECESSITY OF CRYPTANALYTIC EXHAUSTIVE SEARCH

Martin E. Hellman, Ehud D. Karnin and Justin Reyneri

Information Systems Laboratory

Stanford University

Stanford, California 94305

Abstract

It is shown that the amount of computation required for a general cryptanalytic method is equivalent to an exhaustive search over the key space. In particular, any general time-memory tradeoff must do an exhaustive search as a part of the pre- or post-computation.

1. Introduction

In [1] Hellman presented a general cryptanalytic time-memory tradeoff which achieves a significant cost savings compared to exhaustive search of the key space. That time-memory tradeoff requires a precomputation which is equivalent to an exhaustive search, but whose cost can be amortized over many solutions. Hellman conjectured that better time-memory tradeoffs exist, and in particular, that the exhaustive search precomputation might be eliminated. This note shows that the latter conjecture is wrong and that any general cryptanalytic method, including general time-memory tradeoffs, *must* do an exhaustive search, either as part of the pre- or post-computation.

The basic idea behind the proof is surprisingly simple: A general cryptanalytic technique, such as the time-memory tradeoff of [1], applies to any cryptographic system. It must therefore work on any realization of a random cipher,

This work was supported under NSF Grants ENG-10173 and ECS-16161.

as defined in the following section.

2. A random cipher model

Let $C=E(P,K)$ and $P=D(C,K)$ denote the enciphering and deciphering operations involving plaintext P and ciphertext C under key K. When referring to a message which may be either P or C we will use the symbol M. We assume that the number of possible plaintexts, m, equals the number of possible ciphertexts and let $\mathbf{M}=\{1,2,3,\cdots,m\}$ denote the set of possible messages, either plain or ciphertext. Also let $\mathbf{K}=\{1,2,3,\cdots,k\}$ denote the set of possible keys. Then a random cipher is one for which $E(P,K)$, as P ranges over $\mathbf{P}$ with K fixed, is a random permutation of $\mathbf{M}$. Permutations corresponding to different keys are chosen independently, and with the same probability $1/m!$.

Such a random cipher can be thought of as two tables, one for enciphering and one for deciphering. A small example is shown in Figure 1 with $m=6$ messages and $k=4$ keys. Within either table, each column is a random permutation of the elements of $\mathbf{M}$ and any set of columns is statistically independent.

ENCIPHERING TABLE					DECIPHERING TABLE				
	$K=1$	$K=2$	$K=3$	$K=4$		$K=1$	$K=2$	$K=3$	$K=4$
$M=1$	5	4	5	1	$M=1$	2	3	3	1
$M=2$	1	5	2	6	$M=2$	6	4	2	3
$M=3$	3	1	1	2	$M=3$	3	6	5	4
$M=4$	6	2	4	3	$M=4$	5	1	4	6
$M=5$	4	6	3	5	$M=5$	1	2	1	5
$M=6$	2	3	6	4	$M=6$	4	5	6	2

Figure 1

As usual, we assume that the cryptanalyst knows the general system but not the specific key in use. He can therefore encipher or decipher any message under any key of his choosing. In other words, he can ask questions about the system of the form (O,M,K), where O specifies the operation to be performed

($O=E$ to encipher and $O=D$ to decipher), M specifies the plaintext or ciphertext to be operated upon, and K is the key to be used.

We assume that the legitimate users of such a system choose a key, K_0, uniformly from **K** and encipher and decipher messages according to the column indexed by K_0. We allow the cryptanalyst to ask N questions about the system, after which he makes an estimate $\hat{K}$ of K_0. We place no other limits on the computational power of the cryptanalytic algorithm. The estimate $\hat{K}$ is given to an "oracle" which knows K_0 and tells the cryptanalyst whether he is right or wrong. In section 3, we shall establish that an exhaustive search is necessary, by showing that $P(\hat{K}=K_0)<<1$, unless $N \approx k$, the number of keys.

One may argue that it is not necessary for the cryptanalyst to find K_0. Any key which specifies the same permutation as K_0 is just as good for deciphering messages. However, by the "Birthday Problem" [2], the probability of having two identical columns is negligible if $k<<\sqrt{m!}$, a condition which is satisfied by any practical cryptosystem. Keeping in mind this remark, the concept of an "oracle" is analogous to the fact that a guessed key can quickly be checked. For example in a ciphertext only attack [3], a moderate amount of ciphertext can be deciphered and checked for statistical regularity. In a known plaintext or chosen text attack [3], the guessed key $\hat{K}$ is even more easily checked. Note that the proof below applies to all three attacks.

3. The proof

We define G to be the event that the cryptanalyst guesses correctly ($\hat{K}=K_0$), and C to be the event that he accesses the correct column at least once in his N questions. By "accessing the correct column" we mean asking a question (O,M,K) with $K=K_0$, the correct key. As can be seen from the development below, $P(G)$ is usually very close to $P(C)$, so it will prove convenient to condition $P(G)$, as follows,

$$P(G) = P(G|C)P(C) + P(G|C^c)P(C^c). \quad (1)$$

Bounds for the four terms on the right side of (1) are easily obtained:

1) $P(G|C) \leq 1$.

This bound can almost be replaced by an equality if enough information is available to the cryptanalyst to guarantee a unique solution. Shannon's unicity distance [4,5] arguments apply under a ciphertext only attack and can be extended to a known plaintext or chosen text attack, where known plaintext is effectively 100% redundant.

2) $P(C) \leq N/k$.

If the goal were simply to ask one question about the correct column, we could do no better than to access N distinct columns at random. This follows from the assumption that columns are statistically independent -- if we have never accessed the correct column we have no information which would lead us to prefer one of the remaining columns.

3) $P(G|C^c) \leq \frac{1}{k-N}$.

For the same reason as in 2), the cryptanalyst can do no better than to choose randomly from the unaccessed columns if the correct one has not been accessed after N questions. There are at least $k-N$ such columns, more if any column has been accessed more than once.

4) $P(C^c) \leq 1$.

Note that $P(C^c) \approx 1$ if $N << k$ or if a suboptimal strategy is used, e.g., always accessing the same column.

Making the above substitutions, equation (1) becomes

$$P(G) \leq \frac{N}{k} + \frac{1}{k-N}. \quad (2)$$

Both terms on the right side of (2) are small if $N << k$. Thus the cryptanalyst does not have a reasonable chance of correctly estimating K_0 without perform-

ing a significant fraction of an exhaustive search. As a consequence of our result, it becomes evident that any attempt to cryptanalyze a system without performing an exhaustive search must utilize the system's specific structure or low complexity [6] to reduce the amount of computation.

References

[1] M. E. Hellman, "A Cryptanalytic Time-Memory Tradeoff", *IEEE Trans. Inform. Theory*, vol. IT-26, pp. 401-406, July 1980.

[2] W. Feller, *An Introduction to Probability Theory and Its Applications*, 3rd ed. New York: Wiley, 1968.

[3] W. Diffie and M. E. Hellman, "Privacy and Authentication: An Introduction to Cryptography", *Proceedings of the IEEE* vol. 67, No. 3, pp. 397-427, March 1979.

[4] C.E. Shannon, "Communication Theory of Secrecy Systems", *Bell System Tech. J.*, vol. 28, pp. 656-715, Oct 1949

[5] M. E. Hellman, "An Extension to the Shannon Theory Approach to Cryptography", *IEEE Trans. Inform. Theory* , vol IT-23, No. 3, pp. 289-294, May 1977.

[6] G. Chaitin, "On the Difficulty of Computations", *IEEE Trans. Inform. Theory*, vol IT-16, pp. 5-9, Jan. 1970