The Difficulty with Difficulty (A Guide to the Transparencies from the EUROCRYPT '96 IACR Distinguished Lecture)

James L. Massey
Signal & Information Processing Laboratory
Swiss Federal Institute of Technology
ETH Zentrum
CH-8092 Zurich, Switzerland
massey@isi.ee.ethz.ch

July 17, 1996

1 Introduction

I was asked to make the transparencies that I used for my "IACR Distinguished Lecture" at EUROCRYPT '96 available on the IACR web pages. To be useful, I concluded that I would have to provide some explanatory text in which I give some of the explanation that I gave during the oral presentation–thus this paper.

Transparency 2 indicates the kind of provable security that I would like to see eventually reached in cryptography. By way of contrast, transparency 3 indicates the kind of provable security that most people talk about today. From a complexity viewpoint, transparency 2 deals with a *non-uniform* complexity measure while transparency 3 deals with a *uniform* complexity measure (i.e., the same algorithm must compute all instances of the function). In the former case, it makes sense to talk about a function (i.e., one and only one instance of a "function") being difficult, whereas in the latter case one must always talk about the difficulty of an infinite sequence of functions. My lecture was aimed at the former kind of difficulty.

Transparency 3 gives the definition of a one-way function (*a la* Diffie and Hellman). Here I suppose function to mean one and only one "function" so that the underlying notion of difficulty relates to some non-uniform complexity measure. The concept of a one-way function is so fundamental in modern cryptography that it seems to me essential that we face head-on the question of whether such a thing exists.

2 Transposition Complexity

I decided to begin my lecture with a very simple type of complexity, which I called "transposition complexity," for which a complete theory is easily given. The cryptographic puzzle on transparency 5 illustrates this complexity measure.

On transparencies 6 and 7, I review the essentials of Cauchy's theory of permutations on n objects. Cauchy noted that a permutation is fully described by the *cycles* that it induces. Every permutation can be written as the composition of disjoint cycles in which, because of the disjointness, the order of the cycles is arbitrary. A *transposition* is a permutation that swaps two objects but leaves the other $n-2$ objects unchanged. Every permutation can be written as a composition of transpositions. Thus, it seems natural to define the *transposition complexity* of a permutation as the smallest number of transpositions whose composition realizes that permutation.

As shown on transparency 7, an m-cycle can be realized with $m-1$ transpositions. Hence, as shown on transparency 8, a permutation that is the composition of c disjoint cycles has transposition complexity at most $n-c$. It seems almost obvious that $n-c$ is exactly the transposition complexity, but I indicate at the bottom of transparency 8 the questions that must be answered before one can conclude that this is the case. I digressed on transparency 9 to give the simple proof that a *permuation on n objects and its inverse have the same transposition complexity.* Hence, **for the transposition complexity measure, there are no one- way functions among the permutations on n objects.** On transparency 10, I give the proof that a permutation that is the composition of c disjoint cycles has transposition complexity exactly $n-c$. What interests me in this simple proof is its *indirectness.* One shows that composing a transposition with any permutation either increases or decreases the number of disjoint cycles of the latter by exactly 1. Because the identity permutation has n disjoint cycles, it follows that the *inverse* of a c-cycle permuation cannot be realized with fewer than $n-c$ transpositions. Combining this with the equality of transposition complexity for a permutation and its inverse shows that a permutation that is the composition of c disjoint cycles has transposition complexity at least $n-c$, which completes the proof that its transposition complexity is exactly $n-c$.

3 Gate Complexity

Transparency 11 begins the treatment of the specific complexity measure on which I wished to concentrate in my lecture. I begin there by defining a *gate* to be any of the 16 boolean functions of two variables. The *gate complexity* of a boolean function of n variables is then defined on transparency 12 as the smallest number of gates in an acyclic gate network that computes this function.

On transparencies 13 and 14, I use Shannon's famous counting argument [1] for underbounding the gate complexity of the most difficult boolean function of n variables. One simply counts (more precisely, one overbounds) the number of different acyclic gate networks of K gates. If this number is less that the number 2^{2^n} of different boolean functions of n variables, then some such function has gate complexity greater than K. Using apparently very crude bounds, I show on these transparencies that for every $n \geq 8$ there are boolean functions of n variables whose gate complexity is at least $\frac{1}{2}\frac{2^n}{n}$ (and indeed virtually all such functions have at least this gate complexity). Standard synthesis arguments can be used to show that a number of gates greater than this lower bound by only a small factor suffices to realize any such function. On transparency 15, I summarize the upper and lower bounds of this type on gate complexity that were obtained by my former doctoral student, Alain Hiltgen [2]. It seems to me quite remarkable that, for all $n \geq 28$, the most difficult boolean function of n variables has gate complexity that we are sure lies between $\frac{2^n}{n}$ and $2 \times \frac{2^n}{n}$ and that *virtually all* such functions have gate complexity in this same narrow range. It may seem quite surprising then, as mentioned on transparency 16, that no one has exhibited a constructive function of n variables whose gate complexity exceeds $3n - 3$, cf. [3] for this construction. This becomes less surprising when one notes that the strongest general tool available today for proving lower bounds on the gate complexity of a specific boolean function is the quite trivial argument that this complexity must be at least the number of non-idle variables less one.

Transparency 17 shifts the discussion to the gate complexity of *invertible functions* from n bits to n bits, i.e., to functions in the set P_n of permutations on the set $\{0,1\}^n$, which is where cryptographers are especially interested in finding one-way functions (should they exist). On transparency 18, I give the results of an exhaustive search made about six years ago by two of my doctoral students, Alain Hiltgen and Jürg Ganz, for such functions that are more difficult to invert than to compute [4]. There are obviously no such functions in P_1 or in P_2, but we were very surprised to find that there were also none in P_3 even though there are functions in P_3 whose gate complexity is 7. The first such computationally asymmetric function showed up in P_4, requiring 5 gates to compute while its inverse requires 6. This function is given on transparency 18 and is probably the first computationally asymmetric function ever found.

On transparency 19, I briefly consider the more general set B_{nm} of functions from n bits to m bits and mention the almost trivial Lamagne-Savage lower bound [5] on their gate complexity, which is the strongest tool available today for proving lower bounds on the gate complexity of a specific function in B_{nm}. On transparency 20, I show the form of certain binary [i.e., over the field GF(2) of two elements] matrices introduced by Boppana and Lagarios [6]. Alain Hiltgen [7] showed that the linear functions in P_n described by these matrices are *feebly one-way* for all $n \geq 5$ in the sense that the function has gate complexity $n + 1$ while its inverse has gate complexity equal to the integer part of $\frac{3}{2}(n - 1)$. His simple proof is given on transparency 21. What interests me here is that the lower bound on complexity obtained in the proof results from the two rather trivial arguments mentioned above, viz. the idle variable bound and the Lamagna-Savage bound.

Hiltgen [7], with considerably more effort, modified the Boppana-Lagarios matrices to obtain, for every $n \geq 4$, a feebly-one-way function with gate complexity $n+2$ whose inverse has gate complexity $2(n-1)$. These functions in P_n, whose inverse is about twice as complex as the function itself, still hold the world record for computational asymmetry.

On transparency 22, I begin the demonstration that constitutes the climax of this lecture. Again I make use of Shannon's counting argument to show that, for all $n \geq 6$, virtually all functions in P_n have gate complexity greater than $\frac{1}{5}2^n$. Then I overbounded the gate complexity of all functions in P_n simply by applying Hiltgen's upper bound of $2 \times \frac{2^n}{n}$ on the gate complexity of every boolean function of n variables (see transparency 15) to each of the n one-bit output functions of a function in P_n. This gives an upper boundof 2×2^n on the gate complexity of all functions in P_n. But the inverse of a function in P_n is again a function in P_n so this establishes the proposition given on transparency 24, viz.

Proposition: **For all $n \geq 28$, virtually all functions in P_n have gate complexity different by a factor of less than 10 from the gate complexity of their inverse function.**

The inescapable conclusion is that **if there exist functions in P_n that are more than "feebly one-way," then these functions are very rare.**

The final transparency 24 gives the solution of the cryptographic puzzle on transparency 5.

References

[1] C. E. Shannon, "The Synthesis of Two-Terminal Switching Circuits," *Bell System Tech. J.*, vol. 28, pp. 59-98, 1949.

[2] A. P. L. Hiltgen, *Cryptographically Relevant Contributions to Combinatorial Complexity Theory*, (ETH-Zürich Dissertation). Konstanz: Hartung-Gorre Verlag, 1994.

[3] N. Blum, "A Boolean Function Requiring $3n$ Network Size," *Theoretical Comp. Sci.*, vol. 28, pp. 337-345, 1984.

[4] A. P. Hiltgen and J. Ganz, *On the Existence of Specific Complexity–Asymmetric Permutations*, Technical Rept., Signal and Information Proc. Lab., ETH-Zürich, 1992.

[5] E. A. Lamagna and J. E. Savage, "Combinational Complexity of Some Monotone Functions," *Proc. 15th Ann. Symp. Switching and Automata Th.*, pp. 140-144, 1974.

[6] R. B. Boppana and J. C. Lagarias, "One-Way Functions and Circuit Complexity," *Information and Computation*, vol. 74, pp. 226-2240, 1987.

[7] A. P. Hiltgen, "Constructions of Feebly-One-Way Families of Permutations," in *Advances in Cryptology–AUSCRYPT '92* (Eds. J. Seberry and Y. Zheng), Lecture Notes in Computer Science No. 718. New York: Springer, 1993, pp. 422-434.