3.9 Design Criteria for Nonlinear Combiners

From the forgoing discussion we derive design criteria for nonlinear combiners:

- The battery registers should be as long as possible.
- The combining function f should have a low linear potential.

How long should the battery registers be? There are some algorithms for "fast" correlation attacks using the Walsh transformation, in particular against sparse linear feedback functions (that use only a small number of taps) [4]. These don't reduce the complexity class of the attack ("exponential in the length of the shortest register") but reduce the cost by a significant factor. So they are able to attack registers with up to 100 coefficients 1 in the feedback function. As a consequence

- The single LFSRs should have a length of at least 200 bits, and use about 100 taps each.

To assess the number n of LFSRs we bear in mind that the combining function should be "correlation immune", in particular have a low linear potential. A well-chosen Boolean function of 16 variables should suffice, but there are no known recommendations in the literature.

Rueppel found an elegant way out to make the correlation attack break down: Use a "time-dependent" combining function, that is a family $(f_t)_{t\in\mathbb{N}}$. The bit u_t of the key stream is calculated by the function f_t. We won't analyze this approach here.

Observing that the correlation attack needs knowledge of the taps, the security could be somewhat better if the taps are secret. Then the attacker has to perform additional exhaustions that multiply the complexity by factors such as 2^{l_1} for the first LFSR alone. This scenario allows choosing LFSRs of somewhat smaller lengths. But bear in mind that for a hardware implementation the taps are parts of the algorithm, not of the key, that is they are public parameters in the sense of Figure 2.1.

Efficiency

LFSRs and nonlinear combiners allow efficient realizations by special hardware that produces one bit per clock cycle. This rate can be enlarged by parallelization. From this point of view estimating the cost of execution on a usual PC processor is somewhat inadequate. Splitting each of the ≥ 200 bit registers into 4 parts of about 64 bits shifting a single register requires at least 4 clock cycles, summing up to 64 clock cycles for 16 registers. Add some clock cycles for the combining function. Thus one single bit would take about 100 clock cycles. A 2-GHz processor, even with optimized implementation, would produce at most $2 \cdot 10^9/100 = 20$ million bits per second.

As a summary we note:

Using LFSRs and nonlinear combining functions we can build useful and fast random generators, especially in hardware.

Unfortunately there is no satisfying theory for the cryptologic security of this type of random generators, even less a mathematical proof. Security is assessed by plausible criteria that—as for bitblock ciphers—are related to the nonlinearity of Boolean functions.