2.9 Analysis of Congruential Generators with Truncated Output

Cryptanalysis is significantly harder for pseudorandom generators that don't output all bits of the generated numbers. Then the sequence of differences is known at most approximately, greatest common divisors cannot be determined, and the algorithms of Plumstead-Boyar or Boyar/Krawczyk break down.

If the parameters of the pseudorandom generator are known, the cryptanalyst may try an exhaustion. The following consideration lacks mathematical strength. It doesn't presuppose that the pseudorandom generator is linear.

Suppose the generator produces n-bit integers but outputs only q bits (from fixed known positions) and suppresses $n - q$ bits. Then for each q-bit fragment of the output there exist 2^{n-q} possible complete values. In other words, a pseudorandom n-bit integer has the given bits at the given positions with probability $1/2^q$.

To continue we assume for simplicity that the q output bits are the most significant bits. So we decompose the integer x into $x = x_0 2^{n-q} + x_1$ where $0 \le x_1 < 2^{n-q}$. The value x_0, the first q bits, is known. The cryptanalyst runs through the 2^{n-q} different possibilities for x_1. For each choice of x_1 she forms $x = x_0 2^{n-q} + x_1$ and sets $y = s(x)$ with the generating function s of the pseudorandom generator. She compares y with the next q bits of the output that she knows. If the pseudorandom generator is statistically good, then the probability of a hit is $1/2^q$. Thus from the 2^{n-q} test values of x_0 there survive about 2^{n-2q} ones. In the case $q \ge \frac{n}{2}$ she expects exactly one hit. Otherwise she proceeds. After using k substrings of q bits the expected number of hits is about 2^{n-kq}. The expected necessary number of q-bit substrings exceeds k only if $kq \le n$, or $q \ge \frac{n}{k}$. For $q = \frac{1}{4}$ (as in the example $n = 32$, $q = 8$, that is an output of 8 bits of a 32-bit integer) four q-bit fragments suffice (where the exhaustion runs through 2^{24} integers). This trial-and-error procedure is manageable for small modules m. But note that the expense grows exponentially with m (assume the ratio $r = \frac{q}{n}$ of output bits is bounded away from 1).

For linear congruential generators with unknown module Frieze/Kennan/Lagarias, Håstad/Shamir, and J. Stern developed a better (probabilistic) procedure whose first step—finding the module—is summarized in the statement: *The cryptanalyst finds m with high probability if the generator outputs more than $2/5$ of the leading bits.* (without proof).

In the second step the cryptanalyst finds the multiplier a under the assumption that the module m is already known. In the third step she determines the complete integers x_i, or the differences y_i. Also with these

tasks she succeeds except for a negligeable subset of multipliers, and for the "good" multipliers she needs slightly more than one third of the leading bits of x_0, x_1, x_2, and x_3, to derive the complete integers. This enables her to predict all further output of the generator. A similar, somewhat weaker result by J. Stern holds for the case where instead of leading bits the generator outputs "inner bits" of the generated integers.

Thus the cryptanalysis of linear congruential generators reveals fundamental weaknesses, independently of the quality of their statistical properties.

Nevertheless linear congruential generators are useful for statistical applications. It is extremely unlikely that an application procedure "by accident" contains the steps that break a linear congruential generator and reveal its determinism. On the other hand linear congruential generators are disqualified for cryptographic applications once and for all, even with truncated output. However it is an open problem whether the objections also hold for a truncation strategy that outputs "very few" bits, say a quarter (note $\frac{1}{4} < \frac{2}{5}$), or only $\log\log(m)$ bits.

References

- J. Stern: Secret linear congruential generators are not cryptographically secure. FOCS 28 (1987), 421–426.
- Frieze/Håstad/Kannan/Lagarias/Shamir: Reconstructing truncated integer variables satisfying linear congruences. SIAM J. Comput. 17 (1988), 262–280.
- J. Boyar: Inferring sequences produced by a linear congruential generator missing low-order bits. J. Cryptology 1 (1989), 177–184.