SYNCHRONIZATION OF TRULY RANDOM BINARY SEQUENCES

Ingemar Ingemarsson
Dept. of Electrical Engineering
Linköping University
S-581 83 Linköping
SWEDEN

and

James L. Massey
ETH
CH-8092 Zürich
SWITZERLAND

Given a sequence generated by a binary symmetric memoryless source and a delayed version of the same sequence, the problem is to determine the delay. This problem arises for example in direct sequence spread spectrum communication systems where the receiver has to synchronize his added sequence to the sequence used by the transmitter, i.e. he has to determine the delay. The same problem also appears when trying to cryptanalyze the Rip van Winkle cipher, [1].

A straightforward exhaustive search would compare the sequences after having delayed one of them each of the N possible delay values. On the average, two bits are compared before a mismatch is discovered, so the exhaustive method requires on the order of 2N binary comparisons before all but one of the possible delay values are eliminated.

There are, however, more effective procedures to determine the delay. We have proven that at least $\sqrt{N}$ binary comparisons are needed to eliminate all but one of the delay values. We have also constructed an algorithm which requires many fewer binary comparisons than the exhaustive search. The number of comparisons is roughly estimated to be on the order of $\sqrt{N} \cdot \log N$.

The algorithm can be described as the growing of a binary tree of search patterns. At the height h in the tree, the set of search patterns is a subset of the binary words with length h bits. Each of these patterns are used in a search for its first occurrence, if any, in a sebsequence Y of length $[\sqrt{N} + \log_2 N]$ of one of the binary sequences. Each word in the tree is also associated with a set of integers I such that the word appears as a subsequence

$$\{X_{i\sqrt{N}}, X_{1+i\sqrt{N}}, \ldots\}$$

of the other sequence for $i \in I$.

The tree is grown from the root. At each node, the set I is found recursively and the sequence Y is searched as described above. If no match is found, the tree is pruned immediately below that node. Eventually, all branches except one are pruned from the tree. The remaining branch corresponds to a subsequence

$$\{X_{i\sqrt{N}}, X_{1+i\sqrt{N}}, \ldots\}$$

which matches a subsequence of the sequence Y. Since this is the only remaining possible match, the delay has been found.

The main work in the algorithm is done when Y is searched for a given pattern. This step, however, is facilitated by the systematic way the tree is grown. With increasing height in the tree, a decreasingly shorter part of Y has to be searched.

The algorithm has been tested on random and pseudo-random sequences and its performance compared to exhaustive searching. Experimental results will be presented in the talk.

Reference

[1] Massey and Ingemarsson: "The Rip van Winkle Cipher - A Computationally Secure Cipher with Finite Key", submitted to the IEEE International Symposium on Information Theory, 1985.