Abstract—Linear congruential generators (LCGs) of the form \(x_{i+1} = ax_i + b \mod m\), have been used to generate pseudorandom numbers. However these generators have been known to be insecure. This implies that if a small sequence of numbers generated by an LCG is known then it is possible to predict the remaining numbers in the sequence that will be generated. We propose to generate a secure pseudorandom bit sequence by coupling two LCGs as follows. A 1 is output if the first LCG produces an output that is greater than the output of the second LCG and a 0 is output otherwise. The security of this sequence is shown by demonstrating the difficulty of obtaining the initial conditions of the two LCGs given the pseudorandom bit sequence output. If the modulus \(m\) is a power of 2 then efficient circuits can be designed for the proposed generators.

I. INTRODUCTION

Linear congruential generators (LCGs) have been used to generate pseudorandom integer sequences but are known to be cryptographically weak [1], [5], [6], [7], [8], [9]. Coupled chaotic maps have also been used to generate pseudorandom bit sequences [2], [3]. If the chaotic maps are perturbed using linear feedback shift registers then the bit sequences output by coupled chaotic maps are secure. We propose to couple LCGs in a manner that is similar to the coupling of chaotic maps [2]. In this paper we will show that perturbing LCGs is not necessary for security as was the case with chaotic maps. The security of the coupled LCG systems is achieved by the fact that solving inequalities modulo \(m\) requires a search through the entire solution space. The ease of implementation of coupled LCGs makes them an attractive choice compared to other pseudorandom bit generators like the Blum, Blum, Shub generator [4].

A linear congruential generator (LCG) is defined by the recurrence \(x_{i+1} = ax_i + b \mod m\) where \(a, b\) and \(m\) are known and \(x_0\) is secret [1], [11]. The LCG is full period if the period of the sequence generated is \(m\). The LCG has a fixed point (this implies that there exists \(i\) such that \(x_{i+1} = x_i\)) when \((1 - a)^{-1} \mod m\) exists. When this occurs the maximum period of the sequence is \(m - 1\), if the fixed point is not used as an initial condition. The maximum period occurs when the following conditions are satisfied.

1) \(b\) and \(m\) are relatively prime.
2) \((a - 1)\) is divisible by every prime factor of \(m\).
3) \((a - 1)\) is divisible by 4 if 4 divides \(m\).

Shamir and Hastad [7] have shown that it is possible to recover the seed \(x_0\) if at least 1/3 of the leading bits of 3 consecutive numbers in the sequence are known. We shall shortly see that such methods are not applicable to the proposed scheme. A coupled LCG is defined as follows.

\[
x_{i+1} = a_1x_i + b_1 \mod m
\]

\[
y_{i+1} = a_2y_i + b_2 \mod m
\]

\[
B_{i+1} = \begin{cases} 
1 & \text{if } x_{i+1} > y_{i+1} \\
0 & \text{otherwise}
\end{cases}
\]

We assume that \(a_1, b_1, a_2, b_2\) and \(m\) are known and the seed \((x_0, y_0)\) is secret.

Example 1.1: Let \(a_1 = 5, b_1 = 5, a_2 = 5, b_2 = 3,\) and \(m = 8\). Both sequences, \(x_i\) and \(y_i\) have a period of 8 and are hence full period. If the initial condition (or the seed) is \((x_0, y_0) = (2, 7)\), then the sequences are,

\[
\{x_i\} = (7, 0, 5, 6, 3, 4, 1, 2)
\]

\[
\{y_i\} = (6, 1, 0, 3, 2, 5, 4, 7)
\]

The bit sequence \(B_i\) therefore is

\[
\{B_i\} = (1, 0, 1, 1, 0, 0, 0, 0)
\]

We consider the problem of finding the secret if the bit sequence \(B_i\) and the parameters of the coupled LCG system are known. Clearly, since the bit sequence is obtained by comparing the outputs of the LCGs, the method described in Shamir and Hastad [7] will not work. This is because obtaining a third of the bits of \(x_i\) or \(y_i\) is impossible from the bit sequence. The best one could do is to guess these bits such that the inequality imposed on \(x_i\) and \(y_i\) by \(B_i\) is true. If \(m = 2^{100}\), then the number of such guesses for three consecutive outputs is \((2^65)^3\), a prohibitively large number. In the remainder of this paper we will develop an algorithm to find the initial condition for the new coupled LCG described above. We will prove that the developed algorithm has exponential complexity in the size of \(m\) and hence, breaking the coupled LCG system is computationally infeasible. The proof relies on the fact that finding all the solutions to a set of congruential inequalities modulo \(m\) has exponential complexity. In Section II we describe how to find the secret key given the parameters of the coupled LCGs and \(q\) bits of the output sequence.
Theorem 2, is the main result of the paper, and states that $q \approx \log_2 m^2$ to uniquely determine the secret key. In Section III we consider the pseudorandomness properties of the binary sequence output by coupled LCGs. Section IV considers the effectiveness of stream ciphers obtained using coupled LCGs and Section V concludes the paper.

II. Breaking Coupled LCGs

In this section we present an algorithm to determine the initial condition or seed $(x_0, y_0)$ of coupled LCGs with known parameters $a_1, b_1, a_2, b_2,$ and $m$. We assume that $q$ bits of the output of the coupled system are also available. Let these $q$ bits be denoted $(B_1, B_2, \ldots, B_p)$. It is easy to see that the $k^{th}$ output of an LCG $x_{i+1} = ax_i + b \pmod{m}$, is given as,

$$x_k = a^k x_0 + b \sum_{i=0}^{k-1} a^i.$$

This implies that if the $k^{th}$ output of the coupled LCGs is $B_k$, then the following inequalities hold based on whether $B_k$ is 1 or 0.

$$a^k x_0 + b \sum_{i=0}^{k-1} a^i (\mod m) > a^k y_0 + b \sum_{i=0}^{k-1} a^i (\mod m) \quad \text{if } B_k = 1 \quad (4)$$

$$a^k x_0 + b \sum_{i=0}^{k-1} a^i (\mod m) \leq a^k y_0 + b \sum_{i=0}^{k-1} a^i (\mod m) \quad \text{if } B_k = 0 \quad (5)$$

Since $q$ bits of the output $B_k$ are known, we can set up $q$ inequalities $E_k$, $1 \leq k \leq q$, where $E_k$ is an inequality of the form described above.

Example 2.1: For the coupled LCG system of Example 1.1 the inequalities $E_k$, $k = 1, 2, \ldots, 7$ are,

$$5x_0 + 5 > 5y_0 + 3 \pmod{8}$$

$$x_0 + 6 \leq y_0 + 2 \pmod{8}$$

$$5x_0 + 3 > 5y_0 + 5 \pmod{8}$$

$$x_0 + 4 > y_0 + 4 \pmod{8}$$

$$5x_0 + 1 > 5y_0 + 7 \pmod{8}$$

$$x_0 + 2 \leq y_0 + 6 \pmod{8}$$

$$5x_0 + 7 \leq 5y_0 + 1 \pmod{8}$$

Let $S_k$ denote the set of solutions $(x_i, y_i)$ to inequality $E_k$. The intersection of all the sets $S_k$'s for $k \in [1, q]$ gives us a small set of possible values for the seed.

Example 2.2: The solution set $S_1$ to the first inequality is,

$$S_1 = \{(0,0), (0,1), (0,6), (0,3), (0,5), (1,1), (1,6), (2,1), (2,6),$$

$$ (2,3), (2,0), (2,5), (2,2), (2,7), (3,1), (3,6), (3,3), (3,0),$$

$$ (4,1), (5,1), (5,6), (5,3), (5,0), (5,5), (5,2), (6,1), (6,6),$$

$$(6,3)\}$$

The intersection of the solution sets for $E_1$ and $E_2$ (the first two inequalities of Example 2.1) is given by,

$$S_1 \cap S_2 = \{(0,5), (2,0), (2,1), (2,2), (2,3), (2,5), (2,6),$$

$$(2,7), (3,0), (3,1), (3,3), (4,1), (5,1), (5,3), (5,5), (6,3)\}$$

Algorithm 3 computes the intersection of all the $S_k$, $k \in [1, q]$. Algorithm 1 below, which is used by Algorithm 3, finds a set of solutions, $(x, y)$ for the inequality $ax + b \leq cy + d (\mod m)$, where $a, b, c, d \in \mathbb{Z}_m$. Algorithm 2 below, which is used in Algorithm 1, finds a set of solutions to the inequality $ax + b \leq r$, $r \in [0, m - 1]$. To summarize, an inequality in two variables, given by, $ax + b \leq cy + d (\mod m)$, is solved by repeatedly solving, $m$, inequalities in one variable, given by, $ax + b \leq R_i$, which are obtained by substituting $y$ with $i$, where $i$ takes on every value in the range $[0, m - 1]$.

Algorithm 1 Solutions, $(x, y)$ for the inequality $ax + b \leq cy + d (\mod m)$.

Input: inequality $ax + b \leq cy + d (\mod m)$ with $a, b, c, d \in \mathbb{Z}_m$ known.

Output: A solution set $Q_{m-1} = \{(x, y) : x, y \in \mathbb{Z}_m \text{ and } ax + b \leq cy + d (\mod m)$.

for $i = 1 : m - 1$ do

$R_i \leftarrow c \times i + d$

end for

$Q_0 \leftarrow \text{set of solutions}, (x, 0) \text{ for } ax + b \leq R_0 \text{ using Algorithm 2}$

for $i = 1 : m - 1$ do

$Q_i \leftarrow \text{set of solutions}, (x, i) \text{ for } ax + b \leq R_i \text{ using Algorithm 2}$

$Q_i \leftarrow Q_i \cup Q_{i-1}$

end for

$Q_{m-1} \leftarrow \text{set of solutions to } ax + b \leq cy + d (\mod m)$

Algorithm 2 Solutions, $x$ for the inequality $ax + b \leq r$, $r \in [0, m - 1]$.

Input: Inequality $ax + b \leq r$, $r \in [0, m - 1]$.

Output: A solution set $S$ for $x$ in the inequality.

$$S = \emptyset$$

for $i = 0 : r - 1$ do

$x_i \leftarrow a^{-1}(i - b) (\mod m)$

$S \leftarrow S \cup \{x_i\}$

end for

The only way to find solutions to the inequality of Algorithm 1 is to find all possible values of $x$ for every $y$ in $[0, m - 1]$. For every $y$ in the range $[0, m - 1]$, $cy + d$ takes on every value in the range $[0, m - 1]$ if the LCG $y_i = cy + d (\mod m)$ is full period. Let $y$ take on some value $i$ in the range $[0, m - 1]$ and let $R_i = ci + d$, then $ax + b \leq R_i$ implies that $ax + b \equiv r (\mod m)$, where $r = [0, 1, \ldots, R_i]$. Therefore, the inequality $ax + b \leq R_i$ has $R_i + 1$ solutions of the form $x \equiv a^{-1}(r - b)$, $r \in [0, R_i]$. Note that $a^{-1}$ exists because the LCG $x_i = ax_i + b (\mod m)$ is full period. The total number of solutions for the inequality $ax + b \leq cy + d (\mod m)$ is therefore $\sum_{i=0}^{m-1} (R_i + 1) = \sum_{i=0}^{m-1} (i + 1) = \frac{m(m + 1)}{2}$. This is because $R_i$ takes on every value in the range $[0, m - 1]$ (see Lemma 2.1 below).
Lemma 2.1: If two LCGs $x_{i+1} = ax_i + b (mod\ m)$ and $y_{i+1} = cy_i + d (mod\ m)$, have full period then the inequality $ax_i + b \leq cy_i + d (mod\ m)$ has $\frac{m(m+1)}{2}$ solutions for $(x_i,y_i)$.

Corollary 2.1: If two LCGs $x_{i+1} = ax_i + b (mod\ m)$ and $y_{i+1} = cy_i + d (mod\ m)$, have full period then the inequality $ax_i + b > cy_i + d (mod\ m)$ has $\frac{m(m+1)}{2}$ solutions for $(x_i,y_i)$.

We now give a solution to the $q$ inequalities of the form generated by Equations 4 and 5 using bits ($B_1, B_2, ..., B_q$) that are output by the coupled LCGs of Equations 1, 2 and 3. Algorithm 3 below describes how the $q$ inequalities can be solved.

Algorithm 3 Solutions, $(x,y)$ for a set of inequalities of the form $a_i^kx_0 + b_1\sum_{i=0}^{k-1} a_i^j \leq or > a_i^k y_0 + b_2\sum_{i=0}^{k-1} a_i^j (mod\ m)$ for $1 \leq k \leq q$.

Input: Inequalities of the form $a_i^kx_0 + b_1\sum_{i=0}^{k-1} a_i^j \leq or > a_i^k y_0 + b_2\sum_{i=0}^{k-1} a_i^j (mod\ m)$ for $1 \leq k \leq q$.

Output: A solution set $S_q = \{(x,y): x, y \in \mathbb{Z}_m \text{ and all the inequalities are satisfied by (x,y)}.\}

Let inequality $E_i, 1 \leq i \leq q$ denote the $i^{th}$ inequality.

$s_i \leftarrow$ set of solutions, $(x,y)$ for the inequality $E_i$ obtained using Algorithm 1.

for $i = 2 : q$ do
  $s_i \leftarrow$ set of solutions, $(x,y)$ for the inequality $E_i$ obtained using Algorithm 1.
  $s_i \leftarrow s_i \cap s_{i-1}$
end for

$s_q \leftarrow$ set of solutions to the $q$ inequalities.

Theorem 1 below states the number of solutions to two inequalities of the form given as input to Algorithm 1.

Theorem 1: The cardinality of the intersection of the solution sets for $E_i$ and $E_{i+1}$ is approximately $\frac{|S_i|}{2} = \frac{|S_{i+1}|}{2}$, where $|S|$ denotes the cardinality of $S$.

Proof: Without loss of generality let us assume that the bits $B_i$ and $B_{i+1}$ output by the coupled LCGs are 0 and 1 respectively. Then the two inequalities $E_i$ and $E_{i+1}$ are:

$$a_i^kx_0 + b_1\sum_{j=0}^{k-1} a_i^j \leq or > a_i^k y_0 + b_2\sum_{j=0}^{k-1} a_i^j (mod\ m) \quad (6)$$

$$a_i^{k+1}x_0 + b_1\sum_{j=0}^{k} a_i^j > a_i^{k+1} y_0 + b_2\sum_{j=0}^{k} a_i^j (mod\ m) \quad (7)$$

Equation $E_i$ has $\frac{m(m+1)}{2}$ solutions and Equation $E_{i+1}$ has $\frac{m(m-1)}{2}$ solutions. If $m$ is large then the number of solutions to both the equations when solved independently can be approximated by $\frac{m^2}{4}$. Note that the total number of ways of picking a pair $(x,y)$ such that $x, y \in [0, m-1]$ is $m^2$. Therefore any pair $(x,y)$ picked at random will satisfy $E_i$ or $E_{i+1}$ with probability 1/2. If the Equations $E_i$ and $E_{i+1}$ are independent then the probability that a random $(x,y)$ will satisfy both $E_i$ and $E_{i+1}$ is 1/4. Note that the independence assumption is an approximation. Our experiments have shown that this probability is between 1/3 and 2/3. Therefore $|S_i \cap S_{i+1}| = \frac{m^2}{2} = \frac{|S_i|}{2} = \frac{|S_{i+1}|}{2}$.

Corollary 2.2: The cardinality of the intersection of the solution sets of equations $E_1, E_2, ..., E_q$ is approximately $\frac{|S_q|}{2}$, for $1 \leq i \leq q$.

Proof: If $E_1, E_2, ..., E_q$ are independent then any random $(x,y)$ picked uniformly from the $m^2$ possibilities will be a solution of all $q$ inequalities with probability $\frac{1}{2^q}$. This follows from Theorem 1. Therefore,

$$|S_1 \cap S_2 \cap ... \cap S_q| = \frac{m^2}{2^q}.$$ 

Theorem 2: Approximately $[2 \log_2 m]$ consecutive output bits of the coupled LCGs must be known in order to determine a unique seed.

Proof: To obtain a unique solution to the $q$ inequalities $E_1, E_2, ..., E_q$, we must have the number of solutions to the inequalities equal to 1. Therefore $\frac{|S|}{2} = 1$. This implies that $q = 2 \log_2 m$.

From Lemma 2.1 and Theorem 2 we know that in order to solve $q = 2 \log_2 m$ inequalities, approximately $m^{2\log_2 m}$ congruences must be solved to obtain the unique seed. If $m = 2^n$ for some $n$, then this implies that $n^2 2^n$ congruences have to be solved. If $n$ is large, solving this many congruences is computationally infeasible. Thus it is difficult to break the coupled LCGs.

Example 2.3: The solution set $S_1$ from Example 2.2 has cardinality 28 and the solution set $S_1 \cap S_2$ has cardinality 16 that is approximately half of $|S_1|$. This illustrates our claim in Corollary 2.1 and Theorem 1. Note that $q = 2 \log_2 m = 6$, implying that approximately 6 inequalities have to be solved in order to find the secret seed (2,7). We find that at the end of solving 5 inequalities the intersection of sets $S_1$ through $S_5$ is, $S_1 \cap S_2 \cap ... \cap S_5 = \{(2,7)\}$. The number of solutions with each step diminishes as: $28 \rightarrow 16 \rightarrow 6 \rightarrow 3 \rightarrow 1$. This illustrates our claim in Corollary 2.2 and Theorem 2.

Example 2.4: Consider the LCG parameters as $a_1 = 35$, $b_1 = 17$, $c_1 = 45$, $d_1 = 41$, $m = 71$ with a seed of (13,25). Here $q = 2 \log_2 m = 14$ and the algorithm terminates and yields the seed in 14 steps. The number of solutions with each step diminishes as: $2485 \rightarrow 1346 \rightarrow 714 \rightarrow 355 \rightarrow 237 \rightarrow 117 \rightarrow 47 \rightarrow 30 \rightarrow 9 \rightarrow 5 \rightarrow 2 \rightarrow 2 \rightarrow 2 \rightarrow 1$.

III. Pseudorandomness of Coupled LCGs

We applied the NIST pseudorandomness [10] tests for the bits generated by coupled LCGs. These bit sequences were generated for many values of $a_1, b_1, a_2, b_2$ and $m$. We found that LCGs that generate pseudorandom integer sequences also generate pseudorandom bit sequences when they are coupled together, if the number of 1’s and 0’s in the bit sequence are equal. It is also possible to use coupled quadratic congruential generators (QCG) of the form given below, to generate pseudorandom bit sequences.

$$x_{i+1} = a_1x_i^2 + b_1x_i + c_1 (mod\ m) \quad (8)$$

$$y_{i+1} = a_2y_i^2 + b_2y_i + c_2 (mod\ m) \quad (9)$$
Coupled QCGs generate bit sequences that pass all the NIST pseudorandomness tests. The complexity of the algorithms for breaking coupled QCGs is about the same as that for coupled LCGs.

IV. Stream Ciphers

Let the plain text x equal \( x_1 x_2 \ldots x_r \), where \( x_i \in \{0, 1\} \). The cipher text \( y = y_1 y_2 \ldots y_r \), can be obtained from the coupled LCGs as follows: \( y_i = x_i \oplus B_i \). In this case, \( m (m > r) \), the parameters \( a_1, b_1, a_2, b_2 \) and the seed \((x_0, y_0)\) are all secret. This system is insecure if we can predict the bits \( B_i \) output by the coupled LCGs when some of the bits \( B_i \) are known. Coupling of the LCGs, only gives us an inequality between the integer outputs of each LCG. The best one can then do to predict the integer outputs of each individual LCG is to try out all possible values for them. For each such prediction one could then obtain the parameters of each LCG and then predict the binary output of the coupled LCGs. We showed that this is too difficult in the Introduction Section. Another method is to predict the parameters of the individual LCGs and then use the method of this paper to find the secret seed. From the previous sections we know that if we know \( a_1, b_1, a_2, b_2 \) and \( m \), then the probability, that a random \((x, y)\) is the seed, is \( \frac{1}{2^m} \). Clearly this probability is negligible for \( n \) large. If \( a_1, b_1, a_2, b_2 \) and \( m \) are unknown then this probability diminishes further. Therefore any probabilistic algorithm for finding the seed has negligible probability of success.

V. Conclusion

We have shown that coupled LCGs are good candidates for secure pseudorandom bit sequence generation. The weaknesses of a single LCG are removed by the coupling. This is primarily because solving inequalities modulo \( m \) can only be done by searching through the entire solution space. We show that the algorithm to break the coupled LCGs requires the solution of \( n 2^m \) congruences modulo \( m \), where \( m = 2^n \). This is clearly exponential in \( n \) and therefore the task of breaking coupled LCGs is computationally infeasible for large \( m \). We have also shown that at least \( \lceil \log_2 m^2 \rceil \) inequalities have to be solved to obtain a unique seed \((x_0, y_0)\) for the coupled LCGs. Note that lattice based approaches [12] that can be used to solve a system of congruences, are not useful in the present setup because they lead to more algorithmic complexity than the approach in this paper. The reason for this is that solving inequalities requires the repeated solving of congruences thereby leading to more complexity.

REFERENCES