On the evolution of GGHN cipher

Subhadeep Banik, Subhamoy Maitra and Santanu Sarkar

Indian Statistical Institute
Kolkata

December 13, 2011

Outline

Outline

Outline

Outline

GGHN Cipher

- Proposed by Gong et al in 2005
- Modification of RCA with larger state size
- GGHN(n, m) State consists of
  1. A state array S, having 2^n cells
  2. Each cell holds m bit integers
  3. A variable k, of m bits
  4. Index variables i and j

GGHN (n, m) PRGA Algorithm

Input: S-Box S and k from GGHN-KSA(n, m);
Output: m-bit Keystream words z

$i = 0, j = 0, M = 2^m, N = 2^n$;
while *Keystream is generated* **do**
    $i = (i + 1) \bmod N$;
    $j = (j + S[i]) \bmod N$;
1    $k = (k + S[j]) \bmod M$;
    $z = (S[(S[i] + S[j]) \bmod N] + k) \bmod M$;
2    $S[(S[i] + S[j]) \bmod N] = (k + S[i]) \bmod M$;
end

Algorithm 1: GGHN-PRGA(n, m)

Weaknesses

- Once all elements of S and k become even,
  1. It remains in the all even state.
  2. Half of the array locations (those indexed by odd numbers) do not change at all.
- Several short cycles exist.
- For GGHN(n, m), It is possible to
  1. Prove the existence of these cylces.
  2. Construct such cycles by solving modular equations.

A Cycle in GGHN$(2,2)$

Proposition

For the GGHN PRGA$(2,2)$ system, the initial condition $S[0]=1,\ S[1]=1,\ S[2]=0,\ S[3]=1,\ i=j=0,\ k=3$ *forms a cycle of length* 4.

Proof.

The algorithm starts from the given state
$i=0,\ j=0,\ k=3,\ S[0]=1,\ S[1]=1,\ S[2]=0,\ S[3]=1$ and goes through the following states before returning to the original:

$i=1,\ j=1,\ k=0,\ S[0]=1,\ S[1]=1,\ S[2]=1,\ S[3]=1,$
$i=2,\ j=2,\ k=1,\ S[0]=1,\ S[1]=1,\ S[2]=2,\ S[3]=1,$
$i=3,\ j=3,\ k=2,\ S[0]=1,\ S[1]=1,\ S[2]=3,\ S[3]=1.$ □

Remark

$S[i]+S[j]\equiv 2 \bmod 4$ *and* $i=j$ *at all stages.*

A Cycle in GGHN$(2, m)$

Proposition

For the GGHN PRGA$(2, m)$ with $m \geq 2$, consider an initial condition of the form $i = j = 0$, $S[0] = s_0 \equiv 1 \bmod 4$, $S[1] = s_1 \equiv 1 \bmod 4$, $S[2] = s_2 \equiv 0 \bmod 4$, $S[3] = s_3 = 1 \bmod 4$, $k = k_0 \equiv 3 \bmod 4$. If

$$k_0 \equiv -(s_0 + 3s_1 + s_3) \bmod 2^m \text{ and}$$

$$s_2 \equiv -(3s_1 + s_3) \bmod 2^m$$

are satisfied, then a cycle of length 4 will be generated.

Proof

Proof.

The following states occur

$$\begin{pmatrix} i = 1,\ j = 1,\ k = k_0 + s_1 \\ S[0] = s_0,\ S(1) = s_1,\ S[2] = k_0 + 2s_1,\ S[3] = s_3 \end{pmatrix}$$

$$\begin{pmatrix} i = 2,\ j = 2,\ k = 2k_0 + 3s_1 \\ S[0] = s_0,\ S[1] = s_1,\ S[2] = 3k_0 + 5s_1,\ S[3] = s_3 \end{pmatrix}$$

$$\begin{pmatrix} i = 3,\ j = 3,\ k = 2k_0 + 3s_1 + s_3 \\ S[0] = s_0,\ S[1] = s_1,\ S[2] = 2k_0 + 3s_1 + 2s_3,\ S[3] = s_3 \end{pmatrix}$$

$$\begin{pmatrix} i = 0,\ j = 0,\ k = 2k_0 + 3s_1 + s_3 + s_0 \\ S[0] = s_0,\ S[1] = s_1,\ S[2] = 2k_0 + 3s_1 + s_3 + 2s_0,\ S[3] = s_3 \end{pmatrix}$$

□

Proof

Proof.

For this to represent a cycle the conditions $2k_0 + 3s_1 + s_3 + s_0 \equiv k_0 \bmod 2^m$ and $2k_0 + 3s_1 + s_3 + 2s_0 \equiv s_2 \bmod 2^m$ must hold. That is to say k_0 must satisfy the modular equation $k_0 \equiv -(s_0 + 3s_1 + s_3) \bmod 2^m$ and s_2 must satisfy the equation $s_2 \equiv -(3s_1 + s_3) \bmod 2^m$. □

Example

Example

In the system GGHN$(2, 8)$ if we take $s_0 = 69,\ s_1 = 141,\ s_3 = 9$, using the above equations we get $k_0 = 11$ and $s_2 = 80$ and so $i = 0,\ j = 0,\ k = 11,\ S[0] = 69,\ S[1] = 141,\ S[2] = 80,\ S[3] = 9$ forms a cycle of length 4.

Cycles in GGHN(n, m)

Lemma

In the GGHN(n, m) PRGA algorithm, one can obtain a cycle of length 2^n starting with the initial state $i = 0,\ j = 0,\ k = k_0 \equiv -1 \bmod 2^n, S[r] = s_r \equiv 1 \bmod 2^n\ \forall r \in [0, 2^n - 1], r \neq 2,$ and $S[2] = s_2 \equiv 0 \bmod 2^n$ under the conditions

$$k_0 \equiv -\left(s_0 + 3 \cdot s_1 + \sum_{r=3}^{2^n-1} s_r\right) \bmod 2^m \text{ and}$$

$$s_2 \equiv -\left(3 \cdot s_1 + \sum_{r=3}^{2^n-1} s_r\right) \bmod 2^m.$$

Proof

Proof.

The states are evolved as follows

$$\begin{pmatrix} i = j = 1,\ k = k_0 + s_1 \\ S[r] = s_r \forall r \in [0, 2^n - 1], r \neq 2,\ S[2] = k_0 + 2s_1 \end{pmatrix}$$

$$\begin{pmatrix} i = j = 2,\ k = 2k_0 + 3s_1 \\ S[r] = s_r \forall r \in [0, 2^n - 1], r \neq 2,\ S[2] = 3k_0 + 5s_1 \end{pmatrix}$$

$$\begin{pmatrix} i = j = 3,\ k = 2k_0 + 3s_1 + s_3 \\ S[r] = s_r \forall r \in [0, 2^n - 1], r \neq 2,\ S[2] = 2k_0 + 3s_1 + 2s_3 \end{pmatrix}$$

$$\vdots$$

$$\begin{pmatrix} i = j = i_0,\ k = 2k_0 + 3s_1 + \sum_{r=3}^{i_0} s_r,\ S[r] = s_r \\ \forall r \in [0, 2^n - 1], r \neq 2, S[2] = 2k_0 + 3s_1 + \sum_{r=3}^{i_0} s_r + s_{i_0} \end{pmatrix}$$

□

Proof

Proof.

$$\begin{pmatrix} i = j = 0,\ k = 2k_0 + s_0 + 3s_1 + \sum_{r=3}^{2^n-1} s_r,\ S[r] = s_r \\ \forall r \in [0, 2^n - 1], r \neq 2,\ S[2] = 2k_0 + 2s_0 + 3s_1 + \sum_{r=3}^{2^n-1} s_r \end{pmatrix}$$

So we need $k_0 \equiv 2k_0 + s_0 + 3s_1 + \sum_{r=3}^{2^n-1} s_r \bmod 2^m$ and $s_2 \equiv 2k_0 + 2s_0 + 3s_1 + \sum_{r=3}^{2^n-1} s_r \bmod 2^m$. So,

$$k_0 \equiv -\left(s_0 + 3 \cdot s_1 + \sum_{r=3}^{2^n-1} s_r\right) \bmod 2^m \text{ and}$$

$$s_2 \equiv -\left(3 \cdot s_1 + \sum_{r=3}^{2^n-1} s_r\right) \bmod 2^m.$$

□ □

GGHN(n, m)

Example

For example, in GGHN$(8, 32)$ if $s_r = 1 \ \forall \ r \in [0, 2^n - 1]$ except $r = 2$, then we would obtain $k_0 = 2^{32} - (2^8 + 1)$ and $s_2 = 2^{32} - 2^8$.

- A total of $(2^{m-n})^{2^n-1} = 2^{(m-n)(2^n-1)}$ states of this form.
- Cycles of other form may be present.

Randomized variant of the GGHN cipher

- We study a randomized variant of the GGHN cipher
- State Update
  - $k = k + S[\mathbf{j}] \bmod M$
  - $S[\mathbf{S}[\mathbf{i}] + \mathbf{S}[\mathbf{j}] \bmod N] = k + S[\mathbf{i}] \bmod M$
- Present a theoretical model in which indices are chosen independently and uniformly at random in $[0, N-1]$ i.e.
- i, j , $S[i] + S[j] modN$ chosen uniformly random from $[0, N-1]$ in each iteration
- We investigate how long it takes for this model to reach the all zero state.

BIT-RAND GGHN PRGA$(n, 1)$

while *the loop is required to be run* **do**
    Select a, b, c uniformly at random from $[0, N-1]$;
    $k = (k \oplus S[a])$;
    $S[b] = (k \oplus S[c])$;
end

Algorithm 2: BIT-RAND-GGHN-PRGA$(n, 1)$

BIT-RAND GGHN PRGA$(n, 1)$

- We want to find the expected number of iterations required after which all the elements in S as well as k become zero.
- Let q denote the number of 1's in S
- We analyze the Markov chain with state. $(q, k) \in \{0, N\} \times \{0, 1\}$.
- q_t denotes the number of 1's in S after t rounds.
- k_t denotes the value of k after t rounds.
- $(q, k) = (0, 0)$ is the terminating state.

- **while** *the loop is required to be run* **do**
- Select a, b, c uniformly at random from $[0, N-1]$
- $k = (k \oplus S[a])$
- $S[b] = (k \oplus S[c])$

end

$$k_{t+1} = \begin{cases} 1 \oplus k_t, & \text{with probability } \frac{q_t}{N}, \\ k_t, & \text{otherwise} \end{cases} \tag{1}$$

and

$$q_{t+1} = \begin{cases} q_t + 1, & \text{with probability } \frac{q_t}{N}\left(1 - \frac{q_t}{N}\right) \text{ if } k_{t+1} = 0, \\ q_t - 1, & \text{with probability } \frac{q_t}{N}\left(1 - \frac{q_t}{N}\right) \text{ if } k_{t+1} = 0, \\ q_t + 1, & \text{with probability } \left(1 - \frac{q_t}{N}\right)^2 \text{ if } k_{t+1} = 1, \\ q_t - 1, & \text{with probability } \left(\frac{q_t}{N}\right)^2 \text{ if } k_{t+1} = 1, \\ q_t & \text{otherwise.} \end{cases} \tag{2}$$

Markov Chain Matrix

- Define the state chain $A_t \triangleq (q_t, k_t)$ as $0 \triangleq (0,1)$, $1 \triangleq (1,0)$, $2 \triangleq (1,1)$, $3 \triangleq (2,0)$, $4 \triangleq (2,1)$, ..., $2n \triangleq (n,1)$.
- Define the transition matrices M_k,M_q
- $M_k(i,j) = Pr(A_{t+1} = j / A_t = i)$ after change of k
- $M_q(i,j) = Pr(A_{t+1} = j / A_t = i)$ after change of S
- $M = M_q \times M_k$ gives the full transition probabilities after one iteration

RAND GGHN(2,1)

$$M_k = \frac{1}{4}\begin{pmatrix} 4 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 3 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4 & 0 \end{pmatrix}, M_q = \frac{1}{16}\begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 10 & 0 & 4 & 0 & 0 & 0 & 0 & 0 \\ 16 & 0 & 6 & 0 & 4 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 8 & 0 & 3 & 0 & 0 & 0 \\ 0 & 0 & 9 & 0 & 8 & 0 & 9 & 0 & 0 \\ 0 & 0 & 0 & 4 & 0 & 10 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 4 & 0 & 6 & 0 & 16 \\ 0 & 0 & 0 & 0 & 0 & 3 & 0 & 16 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}.$$

RAND GGHN(2,1)

Hence the final transition matrix $M_q M_k$ will be

$$M_f = \frac{1}{64}\begin{pmatrix} 0 & 1 & 3 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 30 & 10 & 8 & 8 & 0 & 0 & 0 & 0 \\ 64 & 6 & 18 & 8 & 8 & 0 & 0 & 0 & 0 \\ 0 & 9 & 3 & 16 & 16 & 3 & 9 & 0 & 0 \\ 0 & 9 & 27 & 16 & 16 & 27 & 9 & 0 & 0 \\ 0 & 0 & 0 & 8 & 8 & 10 & 30 & 0 & 0 \\ 0 & 0 & 0 & 8 & 8 & 18 & 6 & 64 & 0 \\ 0 & 0 & 0 & 0 & 0 & 3 & 9 & 0 & 64 \\ 0 & 0 & 0 & 0 & 0 & 3 & 1 & 0 & 0 \end{pmatrix}.$$

RAND GGHN(2,1)

- To calculate the expected survival time of the process, we need the fundamental matrix $F = (I - M_f)^{-1}$.
- The element F_{ij} gives the expected number of times the process visits state i given the process started from the state j.
- Summing over the columns of F i.e $E = \mathbf{1}^T F$, we get the expected total survival time for each initial state
- Take the dot product $E.v$, where v is the distribution vector for the initial states

RAND GGHN(2,1)

$$F = \begin{pmatrix} 1.33 & 0.25 & 0.33 & 0.29 & 0.29 & 0.29 & 0.29 & 0.29 & 0.29 \\ 5.33 & 5.97 & 5.33 & 5.65 & 5.65 & 5.65 & 5.65 & 5.65 & 5.65 \\ 5.33 & 3.41 & 5.33 & 4.37 & 4.37 & 4.37 & 4.37 & 4.37 & 4.37 \\ 6.00 & 5.10 & 6.00 & 7.80 & 6.80 & 6.99 & 7.11 & 7.11 & 7.11 \\ 10.00 & 8.02 & 10.00 & 10.76 & 11.76 & 11.57 & 11.45 & 11.45 & 11.45 \\ 5.33 & 4.37 & 5.33 & 6.19 & 6.19 & 7.83 & 7.21 & 7.21 & 7.21 \\ 5.33 & 4.37 & 5.33 & 6.19 & 6.19 & 7.01 & 8.03 & 8.03 & 8.03 \\ 1.33 & 1.09 & 1.33 & 1.55 & 1.55 & 1.83 & 1.93 & 2.93 & 2.93 \\ 0.33 & 0.27 & 0.33 & 0.39 & 0.39 & 0.48 & 0.46 & 0.46 & 1.46 \end{pmatrix}.$$

So E, whose entries are sum of columns, will be

$$E = \mathbf{1}^T F = \begin{bmatrix} 40.33 & 32.87 & 39.33 & 43.19 & 43.19 & 46.02 & 46.52 & 47.52 & 48.52 \end{bmatrix}$$

Here the initial state distribution vector $v = \frac{1}{32}(1, 4, 4, 6, 6, 4, 4, 1, 1)$. Hence $E \cdot v = 41.05$ gives the expected number of steps where all elements of S and also k are zero.

Comparison Table

Table: Theoretical bounds and experimental values of $f(N)$ for different values of $N = 2^n$ in BIT-RAND-GGHN-PRGA$(n, 1)$.

N	$f(N)$	Experiment
4	41.05	41.02
8	280.49	279.89
12	1463.27	1469.15
16	7118.88	7111.03
20	33836.28	33433.15
32	3423401.56	-
64	619282894484.52	-
128	14919136419435860915574.98	-
256	6230189288473573925071742121365315064452309.75	-

Conclusions

Corollary

Since $62301892884735739250717421213653150644452309 \approx 2^{142.16}$, *following Table 1, we can say that when length of* S *is 256, all the elements of* S *as well as* k *become zero within expected* $2^{142.16}$ *many steps for* $N = 256$*. For* $RAND - GGHN(8, 32)$ *the expected time to get all elements of* S *and* k *to go to zero is thus* $32 \cdot 2^{142.16} = 2^{147.16}$

Proof.

Each significant bit position can be thought of as a $RAND - GGHN(8, 1)$ system. Multiplying over 32 bit positions we get the result. □

Thank You