TM
[Excerpted from RBS Encryption Patent Application in process]

Appendix 1 - How to generate a stream of random bits

(See DE Knuth "The Art of Computer Programming" (1981) 2:29/31.)

1) Load X

2) Shift X left by one bit

3) If left-bit of 1) was zero then
   goto step 5)
   else
   goto step 4)
   endif

4) XOR X
   with A

   Result

   (XOR means that:
   bits set as 1 in A cause respective bits in X to change state;
   bits set as 0 in A cause respective bits in X not to change state.)

5) Store result in X

6) Output right-most bit of result as a random bit

7) Goto step 1)

The initial, arbitrary state of the machine below is set as follows:
A = '0011' and X = a non-zero binary word, ie, '1011'.

Step No	Binary result (or next step)	Iteration Number														
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		1011	0101	1010	0111	1110	1111	1101	1001	0001	0010	0100	1000	0011	0110	1100
2		0110	1010	0100	1110	1100	1110	1010	0010	0010	0100	1000	0000	0110	1100	1000
3		(4)	(5)	(4)	(5)	(4)	(4)	(4)	(4)	(5)	(5)	(5)	(4)	(5)	(5)	(4)
4	0110		0100		1100	1110	1010	0010					0000			1000
	0011		0011		0011	0011	0011	0011					0011			0011
5		0101	1010	0111	1110	1111	1101	1001	0001	0010	0100	1000	0011	0110	1100	1011
6		1	0	1	0	1	1	1	1	0	0	0	1	0	0	1

NOTE: The right-most '1' in the initial X value of'1011' begins the stream of random bits.

Appendix 2 - Powers of two

```
2**00 =         1
2**01 =         2
2**02 =         4
2**03 =         8
2**04 =        16
2**05 =        32
2**06 =        64
2**07 =       128
2**08 =       256
2**09 =       512
2**10 =      1024
2**11 =      2048
2**12 =      4096
2**13 =      8192
2**14 =     16384
2**15 =     32768
2**16 =     65536
2**17 =   1 31072
2**18 =   2 62144
2**19 =   5 24288
2**20 =  10 48576
2**21 =  20 97152
2**22 =  41 94304
2**23 =  83 88608
```