NUMBER THEORETIC RANDOM BIT GENERATORS
JUNE 8, 2012, 2012

KEVIN LUI AND DEREK HOLLOWOOD

In this paper we discuss 4 number theoretic random bit generators. Some are faster than others, while others are more secure. In the following, we outline the advantages and disadvantages of each RBG, and demonstrate the implementation process. We note that all computations and tests are run on a single core of a Phenom II processor at 2.7 GHz on Linux.

(1) **RSA:** This algorithm takes two primes p and q and the number of bits k desired in the output.
   (a) Security: RSA is very secure. In order to decrypt an RSA bit generator, one must factor the product pq without knowing p and q. Currently there are no known factorisation algorithms which are feasibly efficient to perform this task.
   (b) Efficiency: RSA takes relatively more time to process. At each step, an integer is exponentiated modulo n. Due to the exponentiation by squaring technique, this does not take too much time. However, RSA is more time consuming than other random bit generators (see below).

(2) **Blum Blum Shub:** Like RSA, the Blum Blum Shub algorithm takes two primes p and q and the number of bits k desired in the output. However in this algorithm it is required that p and q be congruent to 3 modulo 4.
   (a) Security: BBS is less secure. It has been shown that decrypting BBS has computational difficulty equivalent to the quadratic resiuosity problem i.e. the problem of determining if an integer is a

square modulo pq where p and q are unknown primes. This is easier than factoring pq because if p and q are known, then an integer x relatively prime to pq is a square modulo pq if and only if

$$x^{(p-1)/2} \equiv 1 \mathrm{mod} p \qquad \text{and} \qquad x^{(q-1)/2} \equiv 1 \mathrm{mod} q$$

(this is easily verified via exponentiation by squaring).

(b) Efficiency: BBS is extremely efficient. Each step of the algorithm involves squaring an integer modulo pq and printing the least significant bit. Computationally this is very simple.

(3) **Dual Elliptic Curve Deterministic Random Bit Generator:** This algorithm is similar to the above algorithms, only instead of using the group $(Z/pqZ)^*$ we use the group of points on the curve

$$y^2 = x^3 + ax + b \ \mathrm{mod} p$$

(plus the point at infinity). So the user inputs a, b, and p to determine the curve and k to specify the number of bits in the output.

(a) Security: The Dual EC DRBG is quite secure. There is no current method known which can decrypt it in reasonable time. As of now it appears that the best way to decrypt the Dual EC DRBG is to solve the discrete logarithm problem (i.e. given elements g and h in a finite cyclic group G, find an integer x so that $g^x = h$). This problem is hypothesised to be in general unsolvable in polynomial time.

(b) Efficiency: This algorithm is less efficient than the others. Adding points on an elliptic curve is time consuming and overall the algorithm requires much more computation.

(4) **Blum-Micali Generator:** The Blum-Micali algorithm takes a large prime p and the number of bit k in the output.

(a) Security: For its simplicity, this algorithm is fairly secure. Decrypting Blum-Micali reduces to solving the discrete logarithm problem mentioned above. For p sufficiently large, this is impossible with today's methods.

(b) Efficiency: The Blum-Micali random bit generator has above average efficiency. While Blum Blum Shub runs the fastest by far, the BM RBG is much quicker than either RSA or the Dual EC DRBG. Most of the run time is spent searching for a primitive root modulo p. The rest of the algorithm consists of simple steps.

It should be noted that the output bit string produced in each of the above algorithms has a high quality of randomness. 0s and 1s are distributed uniformly and bit switches occur about 50 % of the time. In addition, we have performed a poker test (which checks for frequencies of all 32 possible 5-bit strings in the output). The chi-square results are listed below:

Random Bit Generator	χ^2
BBS	31.79
RSA	30.51
Dual EC DRNG	29.69
BM	30.59

Since the average χ^2 of each Random Bit Generator is less than χ^2 with $p = .05$, these generators pass the Poker test.