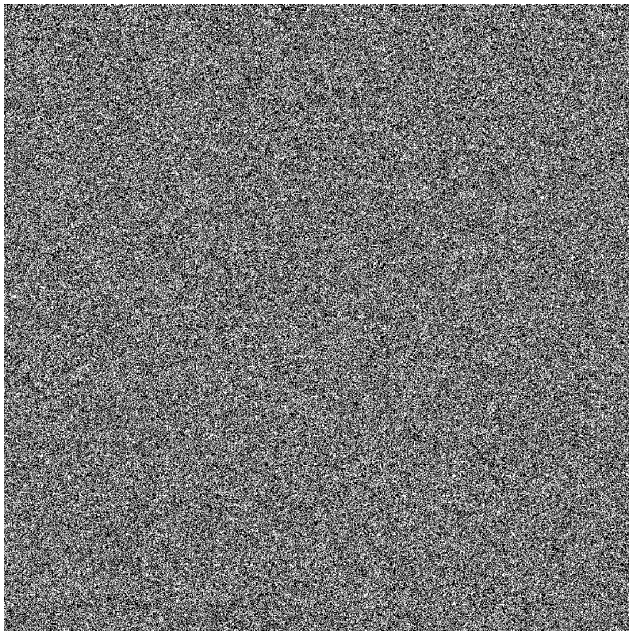
Elliptic Curve DRNGs

Çetin Kaya Koç

koc@cs.ucsb.edu



Elliptic Curve DRNGs

- There are three existing proposals
  - Linear Congruential Generator
  - Power Generator
  - Naor-Reingold Generator
- Some results have been obtained
  - Some complexity results (bounds) for the power generators
  - Studies involving Koblitz curves

Elliptic Curve DRNGs

- Weierstrass form of elliptic curves has been the standard tool
- Interesting applications of character sums, combinatorics, and curves
- Requirement R1 is usually assumed
- Requirement R2: Security proofs of elliptic curve DRNGs are based on the elliptic curve discrete logarithm problem:

$$\boxed{\text{Given } P \text{ and } Q, \text{ compute } d \text{ in } Q = [d]P}$$

Sequence from Points

- Map points $P_n = (x_n, y_n) \in \mathcal{F}_p^2$ into $[0, 1) \times [0, 1)$
- There is a natural map

$$P_n \rightarrow \left(\frac{x_n}{p}, \frac{y_n}{p} \right)$$

  since $\mathcal{F}_p = \{0, 1, \ldots, p-1\}$
- Some applications use only the x coordinate or apply maps to the coordinate values (for example, hash functions or trace maps)

Elliptic Curve Linear Congruential Generator

- For the "initial value" $Q_0 \in E(\mathcal{F}_p)$, consider the sequence

$$Q_k = P \oplus Q_{k-1} = [k]P \oplus Q_0 \quad \text{for} \quad k = 1, 2, \ldots$$

- Easy to construct the following element given two consecutive ones
- Let $Q_k = (x_k, y_k)$ and use $(x_k)_{k=0}$ as sequence in $\mathcal{F}_p$ or normalize to $[0, 1)$ using an enumeration of the field and dividing by p
- Period is linked to the number of points in $\mathcal{E}$
- If the field is $\mathcal{F}_{2^n}$, this sequence is studied

$$\mathrm{Tr}(x_0), \mathrm{Tr}(y_0), \mathrm{Tr}(x_1), \mathrm{Tr}(y_1), \mathrm{Tr}(x_2), \mathrm{Tr}(y_2), \ldots$$

Elliptic Curve Power Generator

- For integer $e \geq 2$, consider the sequence with $Q_0 = P$

$$Q_k = [e]Q_{k-1} = [e^k]P$$

- Determining e from Q_k and Q_{k-1} would be solving the ECDLP
- Constructing the sequence element given longer substrings is related to the generalized ECDH problem

Elliptic Curve Naor-Reingold

- Given an integer vector $A = (a_1, a_2, \ldots, a_n)$, consider the sequence

$$Q_{A,k} = [a_1^{k_1} a_2^{k_2} \cdots a_n^{k_n}]P$$

  where $k = k_1 k_2 \ldots k_n$ is the bit representation of k, $0 \leq k \leq 2^n - 1$
- Ezample: $n = 4$, $l = 19$, and $A = (2, 5, 3, 4)$

$$\begin{aligned}
f_{A,0} &= 2^0 5^0 3^0 4^0 P = P \\
f_{A,1} &= 2^0 5^0 3^0 4^1 P = [4]P \\
f_{A,2} &= 2^0 5^0 3^1 4^0 P = [3]P \\
f_{A,3} &= 2^0 5^0 3^1 4^1 P = [12]P \\
f_{A,11} &= 2^1 5^0 3^1 4^1 P = [24]P = [5]P \\
f_{A,15} &= 2^1 5^1 3^1 4^1 P = [120]P = [6]P
\end{aligned}$$

Research on EC-LCG, EC-PG, and EC-NRG

- Recent work of Tanja Lange, David Kohel, Igor Shparlinski, Berry Schoenmakers, and Vladimir Sidorenko
- Some results of theoretical value; Other results are more practical: If the order of P is at least $p^{0.5+\epsilon}$ then all three sequences are reasonably well distributed

Research on EC-LCG, EC-PG, and EC-NRG

- Cryptanalysis results: A variant of EC-LCG, "dual elliptic curve generator": s_0 random seed, $Q = [a]P$, a is secret

$$\begin{aligned} s_i &= x([s_{i-1}]P) \\ r_i &= \text{lsb}_{240}(x([s_i]Q)) \end{aligned}$$

- r_i are not uniformly distributed; next bit is predictable without knowing a; looks secure if fewer bits are extracted

Dual EC Random Number Generator

- Dual EC RNG is an algorithm to compute pseudorandom numbers starting from some random seed
- It was first time presented in 2004 at a NIST workshop
- Dual EC RNG was standardized by NIST in early 2006, and subsequently appeared in ANSI and ISO standards, among other algorithms to generate deterministic random numbers
- Dual EC RNG was in dozens of commercial cryptographic software libraries
- It was even the default deterministic number generator in RSA Security's BSAFE library

Dual EC RNG Algorithm

- Dual EC RNG algorithm is based on the NIST approved curves with two associated points P and Q on them
- The original standard document, NIST Special Publication 800-90A by the authors E. Barker and J. Kelsey, recommends the use of NIST P-256, P-384, and P-521
- The points P and Q are special points on the curve, generated according to the specification described in the Publication 800-09A

Dual EC RNG Algorithm based on NIST P-256

- The field is GF(p) for $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$
- The elliptic curve $y^2 = x^3 + ax + b$
- The parameters a and b, and the group order n are given as

$$
\begin{aligned}
a &= -3 \bmod p \\
&= 115792089210356248762697446949407573530 \\
&\quad 086143415290314195533631308867097853948 \\
b &= 410583637251521421293261297800472684091 \\
&\quad 144410159937255548352563140394674012 91 \\
n &= 115792089210356248762697446949407573529 \\
&\quad 996955224135760342422259061068512044369
\end{aligned}
$$

Dual EC RNG Algorithm based on NIST P-256

- The Dual EC algorithm uses two points on the curve P and Q whose coordinates are given as

$$
\begin{aligned}
P_x &= 48439561293906451759052585252797914202762949526041747995844080717082404635286 \\
P_y &= 36134250956749795798585127919587881956611106672985015071877198253568414405109 \\
\\
Q_x &= 91120319633256209954638481795610364441930342474826146651283703640232629993874 \\
Q_y &= 80764272623998874743522585409326200078679332703816718187804498579075161456710
\end{aligned}
$$

Dual EC RNG Algorithm based on NIST P-256

- The algorithm starts with a seed $s_0 \in \{0, 1, \ldots, n-1\}$
- Let $\text{LSB}_i(s)$ denote the least significant i bits of s
- For example, $\text{LSB}_3(23) = 7$ since $23 = (10\underline{111})_2$
- The point multiplications are performed using the points P and Q over the curve NIST P-256

input: $s_0 \in \{0, 1, \ldots, n-1\}$ and $k > 0$
output: $240k$ bits
for $i = 1$ to k
    $s_i = x$ coordinate of $[s_{i-1}]P$
    $t_i = x$ coordinate of $[s_i]Q$
    $r_i = \text{LSB}_{240}(t_i)$
return: $r_1, r_2, \ldots, r_k$

Security of the Dual EC RNG Algorithm

- The security of the Dual EC RNG seems to depend on the difficulty of the ECDLP
- Given s_{i-1}, the computation of s_i in "$s_i = x$ coordinate of $[s_{i-1}]P$" is the point multiplication problem: **easy**
- Given s_i, the computation of s_{i-1} in "$s_i = x$ coordinate of $[s_{i-1}]P$" is the elliptic curve discrete logarithm problem: **hard**
- Backtracking (in other words, predicting) is defined as discovering a previous value in the sequence

Security of the Dual EC RNG Algorithm

- Prediction is equivalent to distinguishing the output of the deterministic random number generator from the sequence of uniformly distributed random bits
- It was first shown by Gjosteen and then by Schoenmakers and Sidorenko in 2006 that the output of the Dual EC RNG can be efficiently distinguished from the sequence of uniformly distributed random bits
- The distinguishing attack does not imply solving the ECDLP for the given curve
- It means that the he Dual EC RNG is **insecure** and cannot be used for cryptographic purposes

InSecurity of the Dual EC RNG Algorithm

- Furthermore, Shumow and Ferguson announced in 2007 that there was a "possibility of a back door" in Dual EC
- Shumow and Ferguson explained a way for whoever had generated the special points P and Q to start from one random number produced by Dual EC RNG and predict all subsequent random numbers
- By the end of 2007, in the view of the public cryptographic community, Dual EC RNG was dead and gone

InSecurity of the Dual EC RNG Algorithm

- The media picked up the story in 2013, however, it had a twist :)

  Cryptographers have long suspected that the agency planted vulnerabilities in a standard adopted in 2006 by the NIST and later by the ISO, which has 163 countries as members. Classified NSA memos appear to confirm that the fatal weakness, discovered cryptographers in 2007, was engineered by the agency.

- The surprise for the public cryptographic community was not so much this confirmation of what had already been suspected, but rather that NSA's back-dooring of Dual EC RNG was part of an organized approach to weakening cryptographic standards

InSecurity of the Dual EC RNG Algorithm

- Not mentioned in the reports was the biggest surprise, namely that Dual EC was not dead at all
- A list of "validations" published by NIST showed that Dual EC RNG was provided in dozens of commercial cryptographic software libraries
- Dual EC RNG was even the default pseudorandom number generator in RSA Security's BSAFE library
- Reuters reported that NSA paid RSA $10 million in a deal that set Dual EC RNG as the default method for number generation in the BSAFE library