X9.82 Part 3
Number Theoretic DRBGs

Don B. Johnson
NIST RNG Workshop
July 20, 2004
WHY?

- Asymmetric key operations are about 100 times slower than symmetric key or hash operations.
- Why have 2 DRBGs based on hard problems in number theory?
- Certainly not expected to be chosen for performance reasons!
Some Possible Reasons

- Do not need lots of random bits, but want the potentially **increased assurance**
- Already using an asymmetric key algorithm and want to limit the number of algorithms that IF broken will break my system
- Have an asymmetric algorithm accelerator in the design already
Performance Versus Assurance

- As performance is not likely THE reason an NT DRBG is included in a product
- Make the problem needing to be broken as hard as possible, within reason
- This increases the assurance that the DRBG will not be broken in the future, up to its security level
Quick Elliptic Curve Review

- An elliptic curve is a **cubic equation** in 2 variables X and Y which are elements of a field. If the field is finite, then the elliptic curve is finite
- Point addition is defined to form a group
- ECDLP Hard problem: given $P = nG$, find $n$ where $G$ is generator of EC group and $G$ has order of 160 bits or more
Elliptic Curve \[ y^2 = x^3 + ax + b \]
Toy Example: The Field $\mathbb{Z}_{23}$

- The field $\mathbb{Z}_{23}$ has **23 elements** from 0 to 22
- The “+” operation is addition modulo 23
- The “*” operation is multiplication mod 23
- As 23 is a prime this is a field (acts like rational numbers except it is finite)
The Group $\mathbb{Z}_{23}^*$

- $\mathbb{Z}_{23}^*$ consists of the 22 elements of $\mathbb{Z}_{23}$ excluding 0

<table>
<thead>
<tr>
<th>$5^0$</th>
<th>$5^1$</th>
<th>$5^2$</th>
<th>$5^3$</th>
<th>$5^4$</th>
<th>$5^5$</th>
<th>$5^6$</th>
<th>$5^7$</th>
<th>$5^8$</th>
<th>$5^9$</th>
<th>$5^{10}$</th>
<th>$5^{11}$</th>
<th>$5^{12}$</th>
<th>$5^{13}$</th>
<th>$5^{14}$</th>
<th>$5^{15}$</th>
<th>$5^{16}$</th>
<th>$5^{17}$</th>
<th>$5^{18}$</th>
<th>$5^{19}$</th>
<th>$5^{20}$</th>
<th>$5^{21}$</th>
<th>$5^{22}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>5</td>
<td>2</td>
<td>10</td>
<td>4</td>
<td>20</td>
<td>8</td>
<td>17</td>
<td>16</td>
<td>11</td>
<td>12</td>
<td>13</td>
<td>19</td>
<td>3</td>
<td>15</td>
<td>6</td>
<td>7</td>
<td>14</td>
<td>12</td>
<td>14</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

- The element 5 is called a generator
- The “group operation” is modular multiplication

And return
Solutions to $y^2 = x^3 + x + 1$ Over $\mathbb{Z}_{23}$

<table>
<thead>
<tr>
<th>(0, 1)</th>
<th>(6, 4)</th>
<th>(12, 19)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 22)</td>
<td>(6, 19)</td>
<td>(13, 7)</td>
</tr>
<tr>
<td>(1, 7)</td>
<td>(7, 11)</td>
<td>(13, 16)</td>
</tr>
<tr>
<td>(1, 16)</td>
<td>(7, 12)</td>
<td>(17, 3)</td>
</tr>
<tr>
<td>(3, 10)</td>
<td>(9, 7)</td>
<td>(17, 20)</td>
</tr>
<tr>
<td>(3, 13)</td>
<td>(9, 16)</td>
<td>(18, 3)</td>
</tr>
<tr>
<td>(4, 0)</td>
<td>(11, 3)</td>
<td>(18, 20)</td>
</tr>
<tr>
<td>(5, 4)</td>
<td>(11, 20)</td>
<td>(19, 5)</td>
</tr>
<tr>
<td>(5, 19)</td>
<td>(12, 4)</td>
<td>(19, 18)</td>
</tr>
<tr>
<td>∅</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

There are **28 points** on this toy elliptic curve
ECC DRBG Flowchart
3 Facts and a Question

1. Randomness implies next bit unpredictability
2. The number of points on a curve is approximately the number of field elements
3. All points (X, Y) have a inverse (X, -Y) and at most 3 points are of form (X, 0)

Q: Can I use the X-coordinate of a random point as random bits?
X-Coordinate Not Random

No, I cannot use a raw X-coordinate!
As most X-coordinates are associated with 2 different Y-coordinates, about half the X values have NO point on the curve,
Such X gaps can be considered randomly distributed on X-axis
Look at toy example to see what is going on
Toy Example of X Gaps

Possible X coordinate values: 0 to 22
X values appearing once: 4
Twice: 0, 1, 3, 5, 6, 7, 9, 11, 12, 13, 17, 18, 19
None: 2, 8, 10, 14, 15, 16, 20, 21, 22
An X coordinate in bits from 00000 to 10110
If I get first 4 bits of X value of 0100a, I know a must be a 1, as 9 exists but 8 does not
1-bit Predictability

- If output 4 bits as a random number, the next bit is **completely predictable**!
- This property also holds for 2-bit gaps, 3-bit gaps, etc. with decreasing frequency.
- **Bad luck is not an excuse** for an RBG to be predictable!
- The solution: **Truncate** the X-coordinate. Do not give all that info out. How much?
## X Coordinate Truncation Table

<table>
<thead>
<tr>
<th>Field Type</th>
<th>Truncate Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prime field</td>
<td>Truncate at least 13 leftmost bits of x coordinate</td>
</tr>
<tr>
<td>Binary Field, cofactor = 2</td>
<td>Truncate at least 14 leftmost bits of x coordinate</td>
</tr>
<tr>
<td>Binary Field, cofactor = 4</td>
<td>Truncate at least 15 leftmost bits of x coordinate</td>
</tr>
</tbody>
</table>
Truncation

- This truncation will ensure no bias greater than $2^{-44}$
- Reseed every 10,000 iterations so bias effect is negligible
- To work with bytes, round up so remainder of X-coordinate is a multiple of 8 bits, this truncates from 16 to 19 bits
Quick RSA Review

- Choose odd public exponent e and primes p and q such that e has no common factor with p or q, set n = pq
- Find d such ed = 1 mod (p-1)(q-1)
- Public key is (e, n), private key is (d, n)
- Hard to find d from (e, n) if n >= 1024 bits
- \((M^e \mod n)\) is **hard to invert** for most M
Micali-Schnorr DRBG

\[ y_i = s^e \mod n \]

\[ r = \text{leftmost } x \text{ bits} \]

\[ r = \text{rightmost } \log(n) \times x \text{ bits} \]

\[ \text{pseudorandom bits} \]
Unlooped Flowchart

- $S_0$ → $S_1$ → $S_2$ →
  - $R_1$
  - $R_2$
Micali-Schnorr Truncation

- For MS truncation, we only use the RSA **hard core bits** as random bits
- This has high assurance that the number theory problem to be solved is as hard as possible!
- Reseed after 50,000 iterations
### NIST/ANSI X9 Security Levels Table

<table>
<thead>
<tr>
<th>Security Levels (in bits)</th>
<th>ECC (order of G in bits)</th>
<th>MS (RSA) (modulus in bits)</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>160</td>
<td>1024, 10 hardcore bits</td>
</tr>
<tr>
<td>112</td>
<td>224</td>
<td>2048, 11 hardcore bits</td>
</tr>
<tr>
<td>128</td>
<td>256</td>
<td>3072, 11 hardcore bits</td>
</tr>
<tr>
<td>192</td>
<td>384</td>
<td>Not specified</td>
</tr>
</tbody>
</table>
Number Theory DRBGs

Summary

- 2 Number Theory DRBGs are specified based on **ECC** and **RSA**
- Use one for **increased assurance**, but do not expect it to be the fastest one possible
- No one has yet asked for an FFC DRBG, straightforward to design from ECC DRBG, but specifying algorithm and validation method has a cost