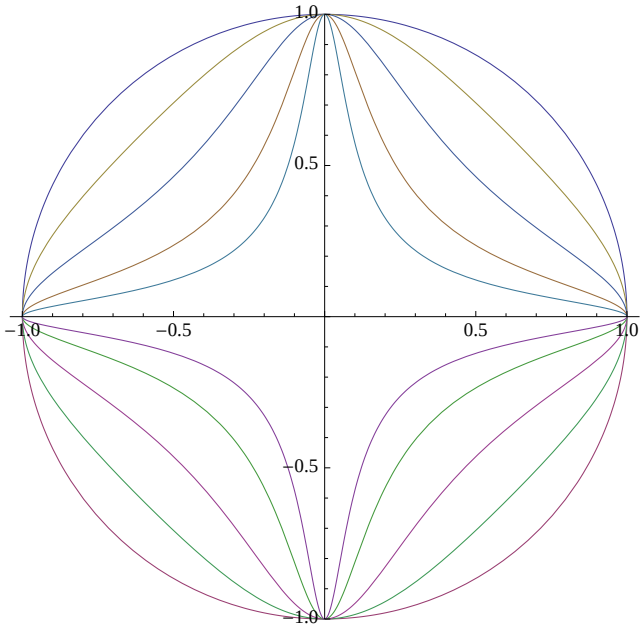
True & Deterministic Random Number Generators

Çetin Kaya Koç

http://cs.ucsb.edu/~koc

koc@cs.ucsb.edu



Random Numbers in Cryptography

- Session keys
- Signature keys and parameters
- Authentication protocols
- Ephemeral keys (DSA, ECDSA, ElGamal)
- Zero-knowledge protocols
- IVs for block ciphers
- Blinding and masking values
- ...

Properties of Random Numbers

Silent Requirement: The random numbers should assume all admissible values with equal probability and should be independent from predecessors and successors.

Properties of Random Numbers

Silent Requirement: The random numbers should assume all admissible values with equal probability and should be independent from predecessors and successors.

This characterizes an **ideal random number generator**

Ideal RNGs

- Even with maximal knowledge and unlimited computational power an attacker has no better strategy than "blind" guessing
  $\rightarrow$ Brute force attack

Ideal RNGs

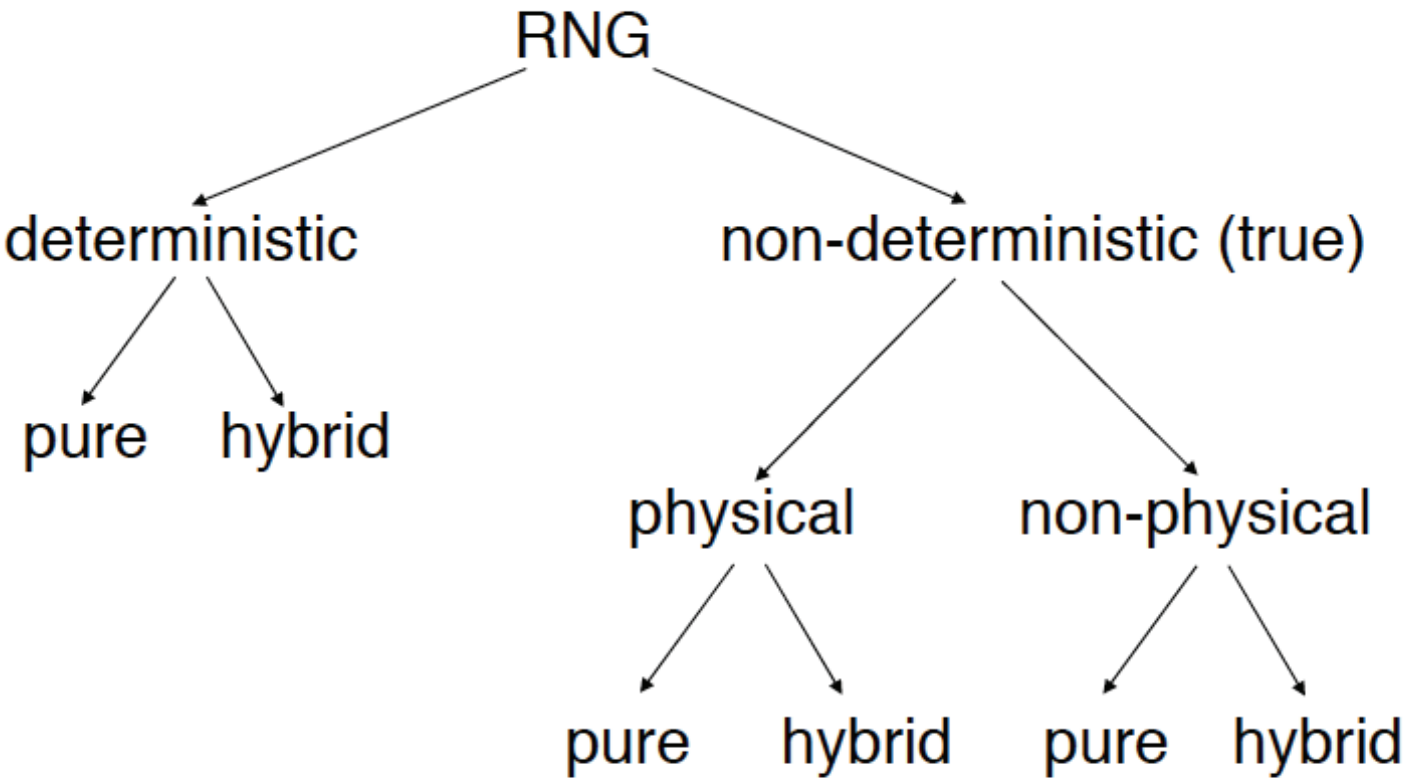
- Even with maximal knowledge and unlimited computational power an attacker has no better strategy than "blind" guessing
  $\rightarrow$ Brute force attack
- Guessing n random bits costs 2^{n-1} trials in average
- The guess work remains invariant in the course of time
  $\rightarrow$ Today, in 2 years, in 100 years

Ideal RNGs

- Even with maximal knowledge and unlimited computational power an attacker has no better strategy than "blind" guessing
  $\rightarrow$ Brute force attack
- Guessing n random bits costs 2^{n-1} trials in average
- The guess work remains invariant in the course of time
  $\rightarrow$ Today, in 2 years, in 100 years

- However, An ideal RNG is a mathematical construct!

Real World RNGs



Random Number Generators in Cryptography

- Deterministic RNGs are also known as pseudorandom number generators

Random Number Generators in Cryptography

- Deterministic RNGs are also known as pseudorandom number generators
- Hybrid deterministic RNGs and hybrid true RNGs apply design criteria from both deterministic RNGs and non-deterministic RNGs

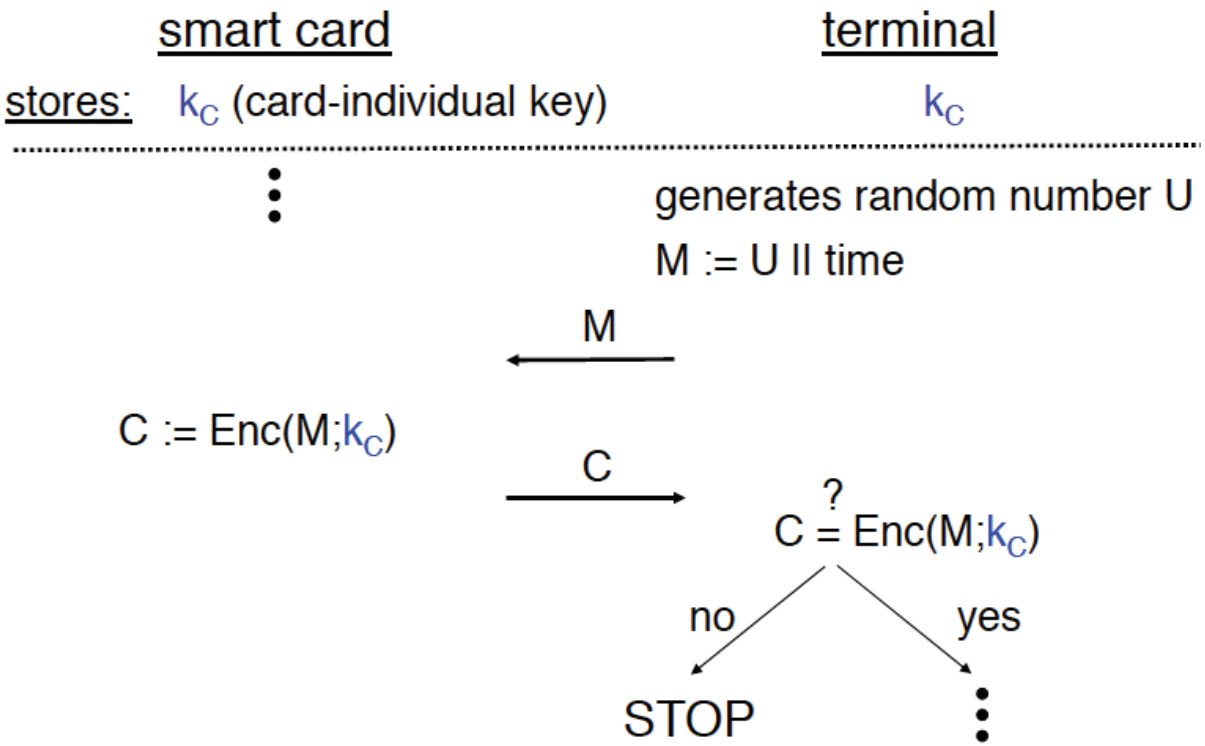
Random Number Generators in Cryptography

- Deterministic RNGs are also known as pseudorandom number generators
- Hybrid deterministic RNGs and hybrid true RNGs apply design criteria from both deterministic RNGs and non-deterministic RNGs
- True random numbers cannot be computed on deterministic computers, they are best produced using physical RNGs which operate by measuring a well controlled and specially prepared random physical process

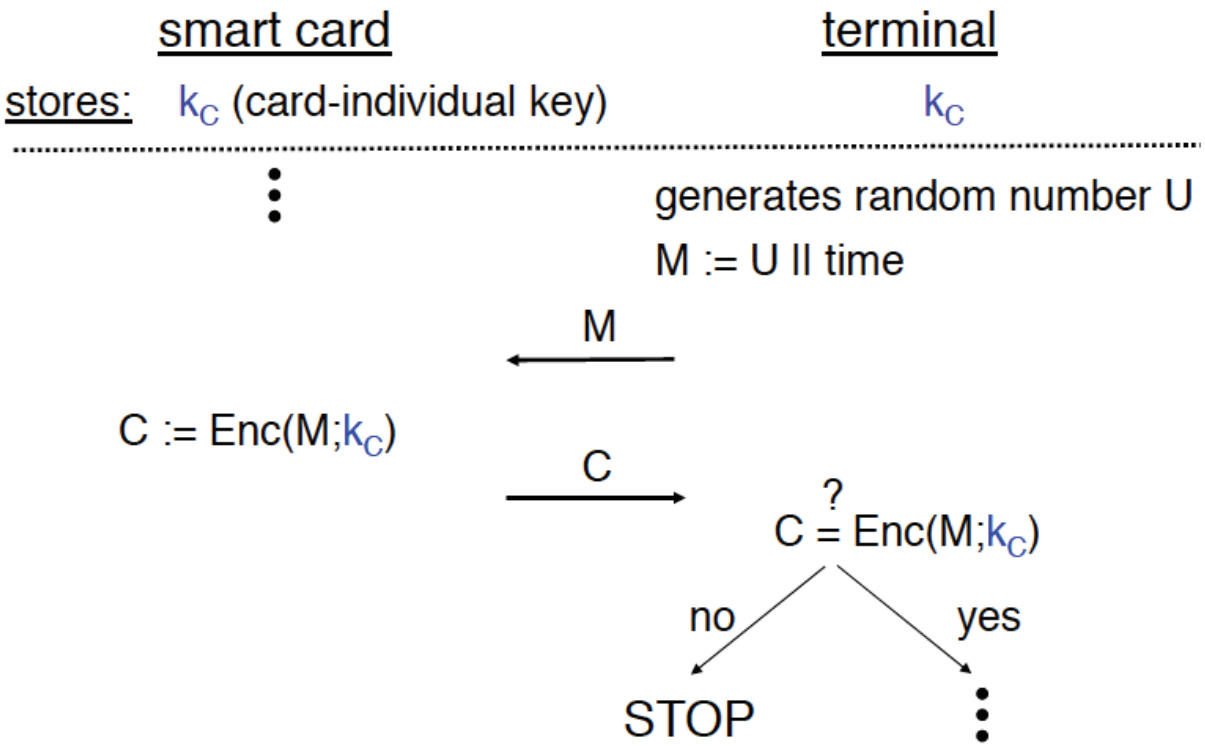
Random Number Generators in Cryptography

- Deterministic RNGs are also known as pseudorandom number generators
- Hybrid deterministic RNGs and hybrid true RNGs apply design criteria from both deterministic RNGs and non-deterministic RNGs
- True random numbers cannot be computed on deterministic computers, they are best produced using physical RNGs which operate by measuring a well controlled and specially prepared random physical process
- Especially valuable are *information-theoretic provable RNGs* which, at state of the art, seem to be possible only by exploiting randomness inherent to certain quantum systems

Challenge-Response Protocol



Challenge-Response Protocol



To prevent replay attacks the random numbers $U_1, U_2, \ldots$ should be distinct with overwhelming probability

Security Requirement R1

- *Random numbers should not show any statistical weaknesses*

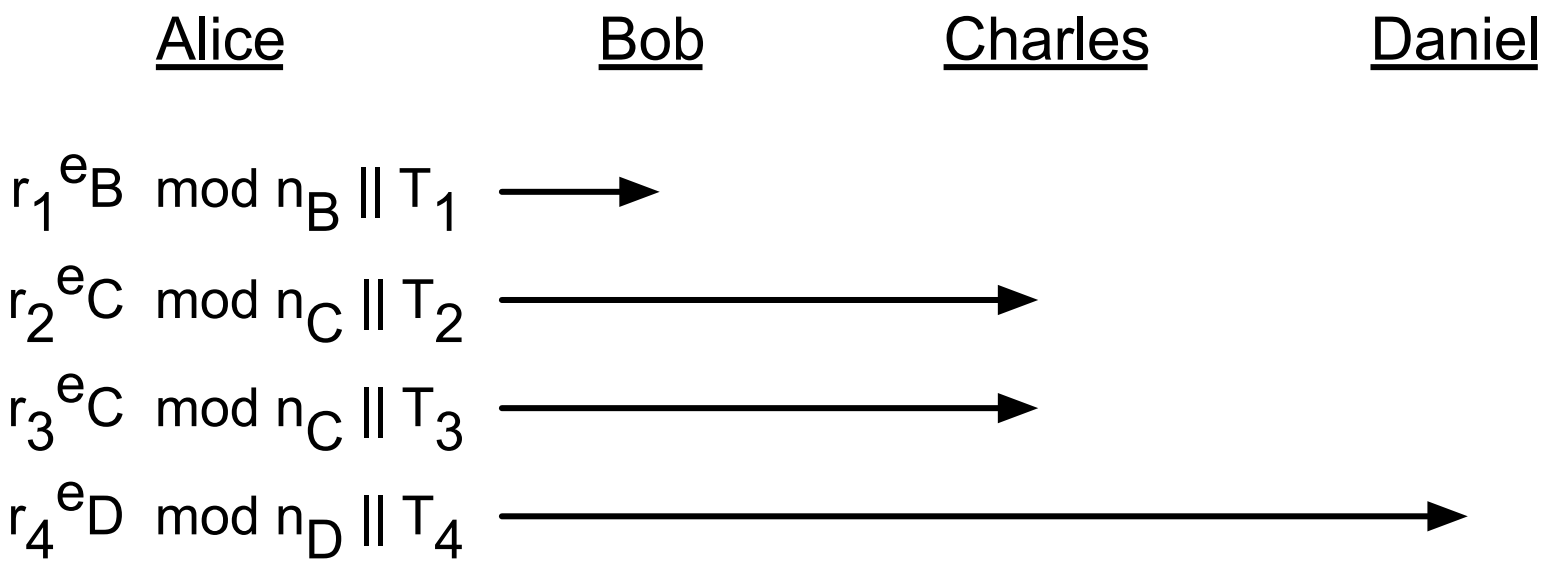
Security Requirement R1

- *Random numbers should not show any statistical weaknesses*
- Requirement R1 is usually verified by statistical tests

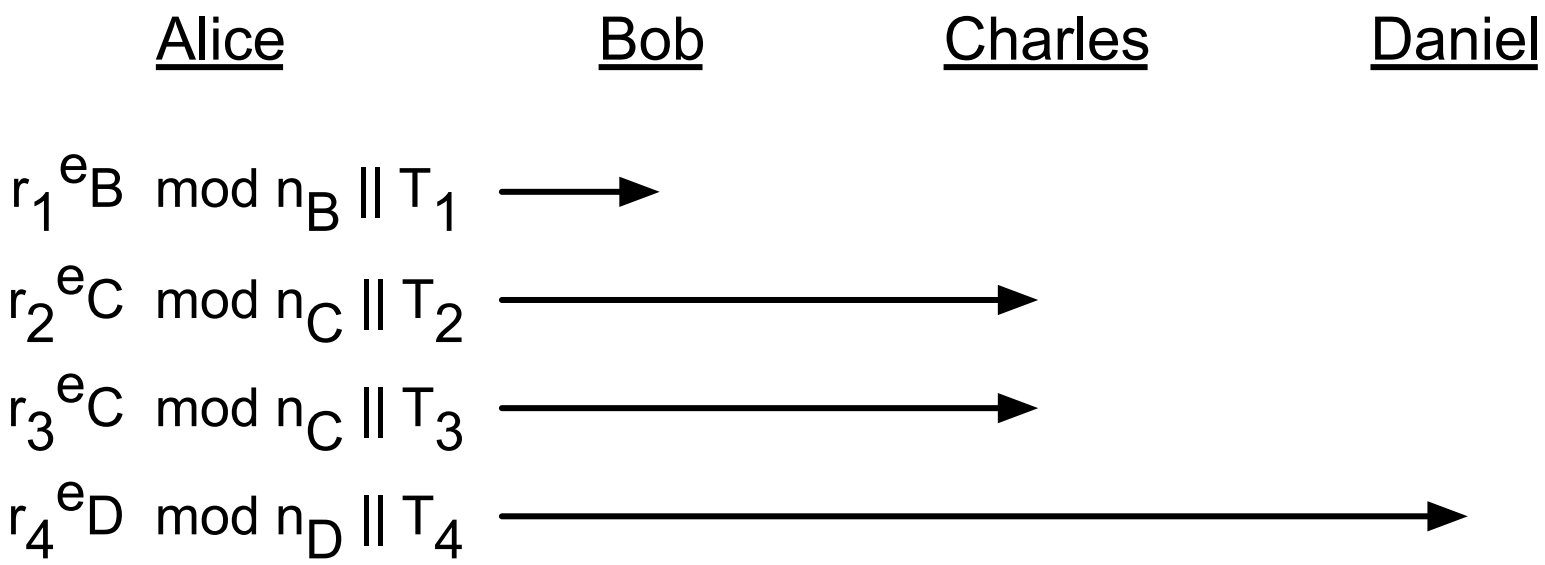
Security Requirement R1

- *Random numbers should not show any statistical weaknesses*
- Requirement R1 is usually verified by statistical tests
- Is Requirement R1 is sufficient?

Key Exchange Protocol



Key Exchange Protocol



Privileged attacker Charles:
The knowledge of r_2 and r_3 may allow him to guess r_3

Security Requirement R2

- *The knowledge of subsequences of random numbers should not allow one to compute predecessors or successors practically or to guess them with non-negligibly larger probability than without knowledge of these subsequences*

Security Requirement R2

- *The knowledge of subsequences of random numbers should not allow one to compute predecessors or successors practically or to guess them with non-negligibly larger probability than without knowledge of these subsequences*
- Requirement R2 implies backward and forward security

Security Requirement R2

- *The knowledge of subsequences of random numbers should not allow one to compute predecessors or successors practically or to guess them with non-negligibly larger probability than without knowledge of these subsequences*
- Requirement R2 implies backward and forward security
- Requirement R2 can be thought of as the union of R3 (backward security) and R4 (forward security)

Security Requirements

- The minimum requirements on the random numbers depend on the intended applications

Security Requirements

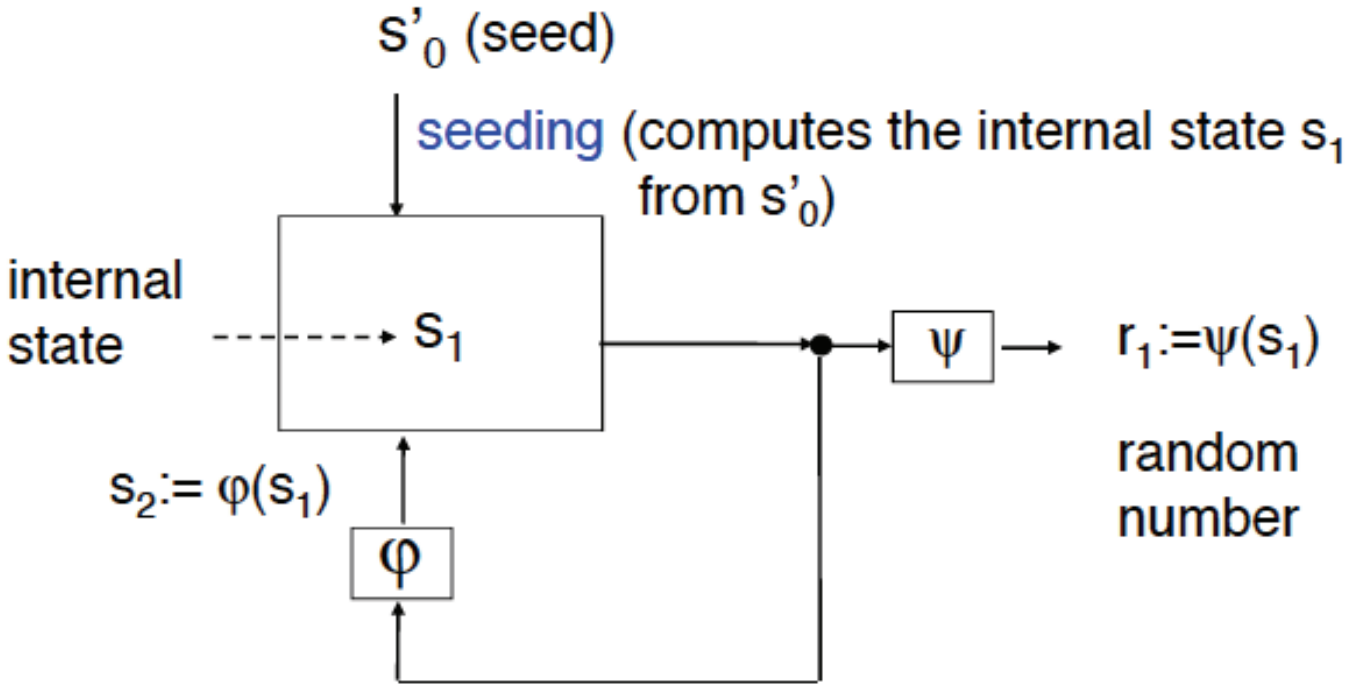
- The minimum requirements on the random numbers depend on the intended applications
- For sensitive applications, e.g., the generation of session keys or signature parameters, Requirement R2 is indispensable

DRNGs

- A pure DRNG starts with a seed (s_0') value and using a "seeding procedure" computes the first internal state s_1 from s_0'
- The output is the random number r_1 which is computed using the output function Ψ while the next state s_2 is computed using the state transition function Φ as

$$\begin{aligned} s_1 &= \textit{seeding}\ (s_0') \\ r_1 &= \Psi(s_1) \\ s_2 &= \Phi(s_1) \end{aligned}$$

Pure DRNG Schematic



φ : state transition function

ψ : output function

DRNG Examples

- Linear Feedback Shift Registers (LFSRs)
- Cellular Automata (CA)
- Linear Congruential Generators (LCGs)
- Block cipher based methods
- Hash function based methods
- Number-theoretical methods: Blum-Blum-Shub, RSA, Rabin
- Elliptic curve methods: LCG, Power Generator, Naor-Reingold
- New: Edward curves method

DRNG Advantages

DRNGs as random number generators have many advantages:

- low cost
- no dedicated hardware is required
- implementations can be done in software
- identical seed values imply identical random numbers which is a necessary condition for using them as stream ciphers

DRNG Disadvantages

However, there are disadvantages:

- For pure DRNGs, the output is completely determined by the seed
- Output sequences of pure DRNGs cannot be truly independent
- They may behave as output sequence of an ideal RNG at most with respect to certain aspects
- The internal state has to be protected even if the device is not active

Security Requirements of DRNGs

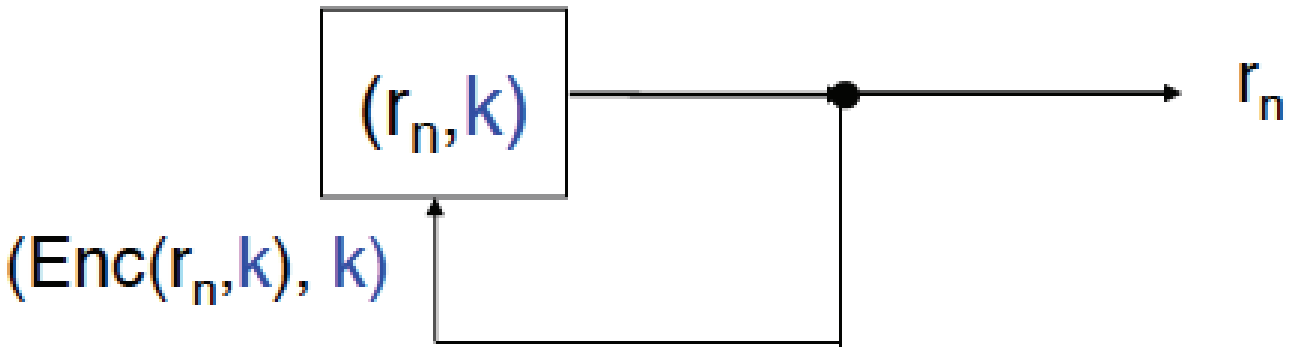
- LFSRs: usually meet Requirement R1, but they do not meet R2
- CA: on certain conditions meet R1 and R2
- LCGs: usually meet Requirement R1, on certain conditions meet R2

Security Requirements of DRNGs

- LFSRs: usually meet Requirement R1, but they do not meet R2
- CA: on certain conditions meet R1 and R2
- LCGs: usually meet Requirement R1, on certain conditions meet R2
- Simple structures are useful for efficient implementations but they have serious security shortcomings

Block Cipher DRNGs

- Block cipher based DRNGs: k key (to be kept secret) and internal state $s_n = (r_n, k)$ and $s_{n+1} = (E_k(r_n), k) = (r_{n+1}, k)$



- Internal state: $s_n = (r_n, k)$
- $s_{n+1} = (E(r_n, k), k) = (r_{n+1}, k)$

Block Cipher DRNGs

- Block cipher based DRNGs meet R1: ciphertext from a strong cipher should not have any statistical weakness

Block Cipher DRNGs

- Block cipher based DRNGs meet R1: ciphertext from a strong cipher should not have any statistical weakness
- They also meet R2: only if the encryption and decryption functions are secure against chosen-plaintext attacks

Block Cipher DRNGs

- Block cipher based DRNGs meet R1: ciphertext from a strong cipher should not have any statistical weakness
- They also meet R2: only if the encryption and decryption functions are secure against chosen-plaintext attacks
- This "security proof" is typical for DRNGs: tracing back to the recognized properties of well-known cryptographic primitives

Block Cipher DRNGs

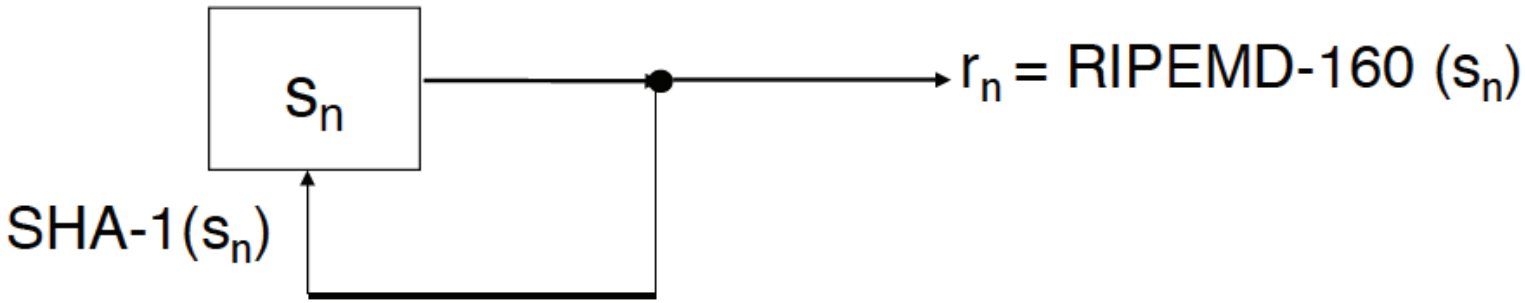
- Block cipher based DRNGs meet R1: ciphertext from a strong cipher should not have any statistical weakness
- They also meet R2: only if the encryption and decryption functions are secure against chosen-plaintext attacks
- This "security proof" is typical for DRNGs: tracing back to the recognized properties of well-known cryptographic primitives
- For AES and Triple-DES encryption functions, may assume that this DRNG meets R2

Block Cipher DRNGs

- Block cipher based DRNGs meet R1: ciphertext from a strong cipher should not have any statistical weakness
- They also meet R2: only if the encryption and decryption functions are secure against chosen-plaintext attacks
- This "security proof" is typical for DRNGs: tracing back to the recognized properties of well-known cryptographic primitives
- For AES and Triple-DES encryption functions, may assume that this DRNG meets R2
- In the 80s, the same conclusion was justified for Single-DES, but this conclusion is no longer valid!

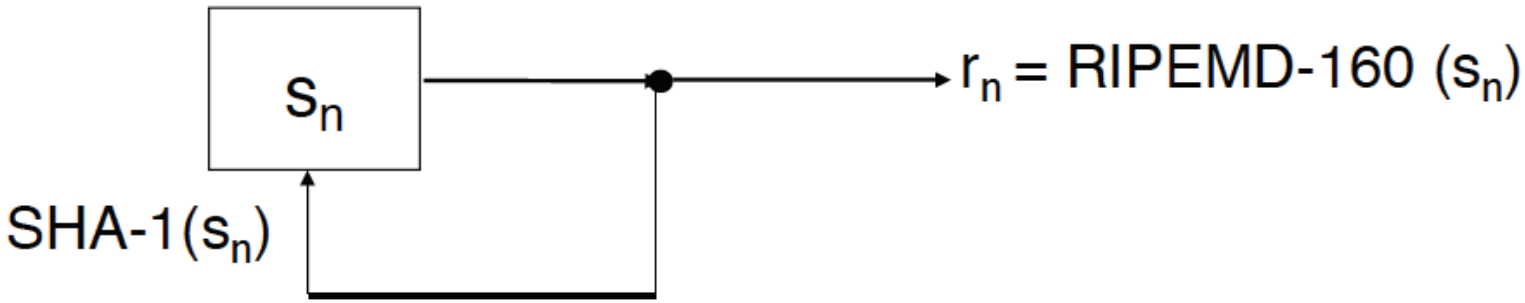
Hash Function DRNGs

- s_n is a 160-bit vector



Hash Function DRNGs

- s_n is a 160-bit vector



- Fulfills R1 and R2 (both backward and forward security)

Cryptographically Secure DRNGs

- Their security is based on intractability assumptions, e.g., factoring is hard or DLP is hard
- Examples: Blum-Blum-Shub, RSA, Rabin bit generators

Cryptographically Secure DRNGs

- Their security is based on intractability assumptions, e.g., factoring is hard or DLP is hard
- Examples: Blum-Blum-Shub, RSA, Rabin bit generators
- On the basis of these intractability assumptions certain security properties can be proved, such as next-bit security
- Usually only asymptotic security properties can be proved, for example, with increasing RSA modulus

Cryptographically Secure DRNGs

- Their security is based on intractability assumptions, e.g., factoring is hard or DLP is hard
- Examples: Blum-Blum-Shub, RSA, Rabin bit generators
- On the basis of these intractability assumptions certain security properties can be proved, such as next-bit security
- Usually only asymptotic security properties can be proved, for example, with increasing RSA modulus
- Not used in practice due to their low output rate

Security Requirement R4

- For specific applications, Requirement R4 (enhanced forward security) is desirable

Security Requirement R4

- For specific applications, Requirement R4 (enhanced forward security) is desirable
- *It should not be practically feasible to compute future random numbers from the internal state or to guess them with non-negligibly larger probability than without the knowledge of the internal state*

Security Requirement R4

- For specific applications, Requirement R4 (enhanced forward security) is desirable
- *It should not be practically feasible to compute future random numbers from the internal state or to guess them with non-negligibly larger probability than without the knowledge of the internal state*
- **Attack Scenario:** An attacker is able to read or to manipulate the internal state of a DRNG without being noticed by the user/owner of the DRNG who uses the subsequent random numbers

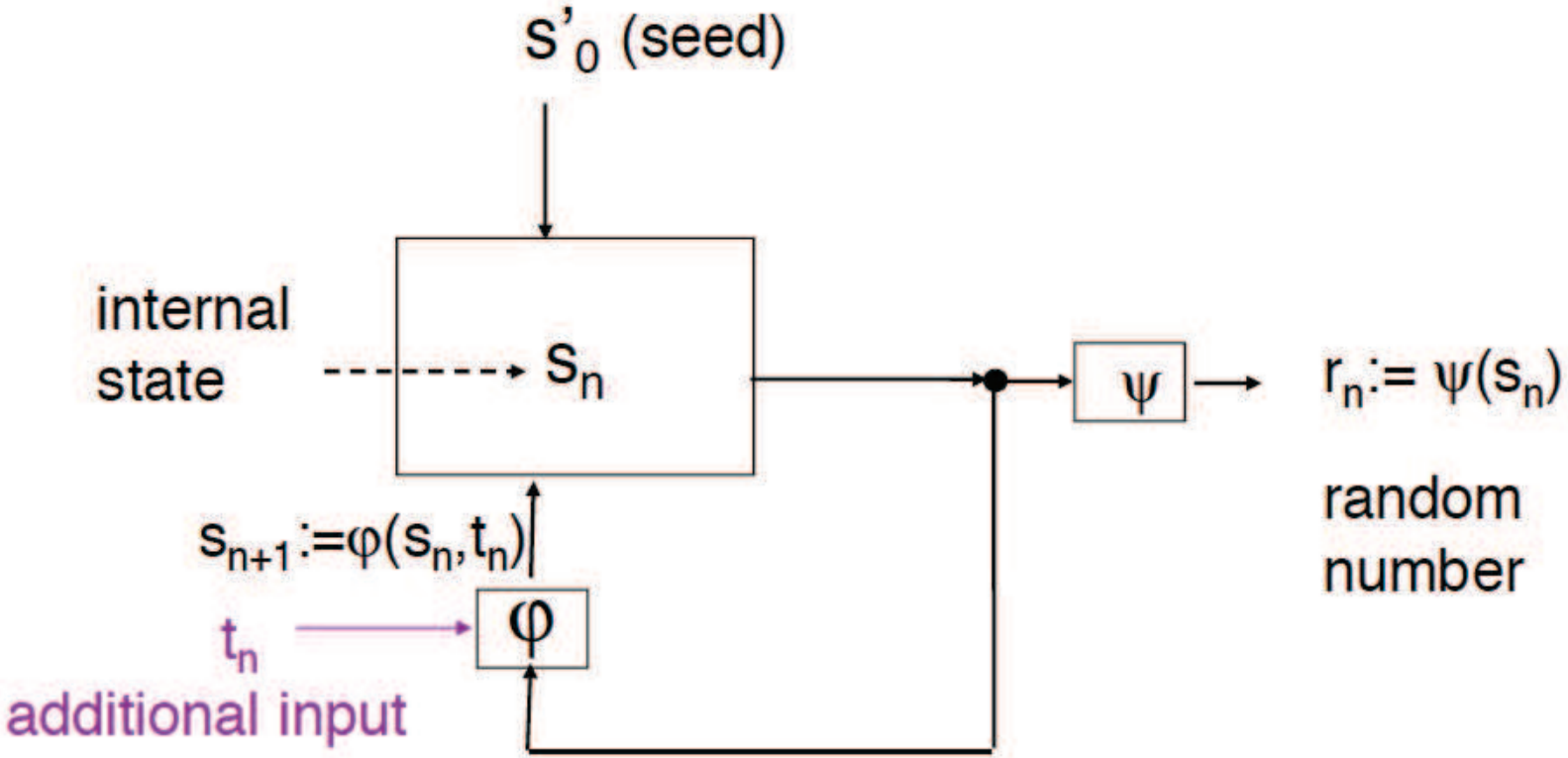
Security Requirement R4

- For specific applications, Requirement R4 (enhanced forward security) is desirable
- *It should not be practically feasible to compute future random numbers from the internal state or to guess them with non-negligibly larger probability than without the knowledge of the internal state*
- **Attack Scenario:** An attacker is able to read or to manipulate the internal state of a DRNG without being noticed by the user/owner of the DRNG who uses the subsequent random numbers
- Pure DRNGs cannot fulfill Requirement R4

Security Requirement R4

- For specific applications, Requirement R4 (enhanced forward security) is desirable
- *It should not be practically feasible to compute future random numbers from the internal state or to guess them with non-negligibly larger probability than without the knowledge of the internal state*
- **Attack Scenario:** An attacker is able to read or to manipulate the internal state of a DRNG without being noticed by the user/owner of the DRNG who uses the subsequent random numbers
- Pure DRNGs cannot fulfill Requirement R4
- A hybrid DRNG may fulfill R4 for the random numbers that are generated after the first update of its internal state with random data after the internal state has been compromised

Hybrid DRNGs



Hybrid DRNGs

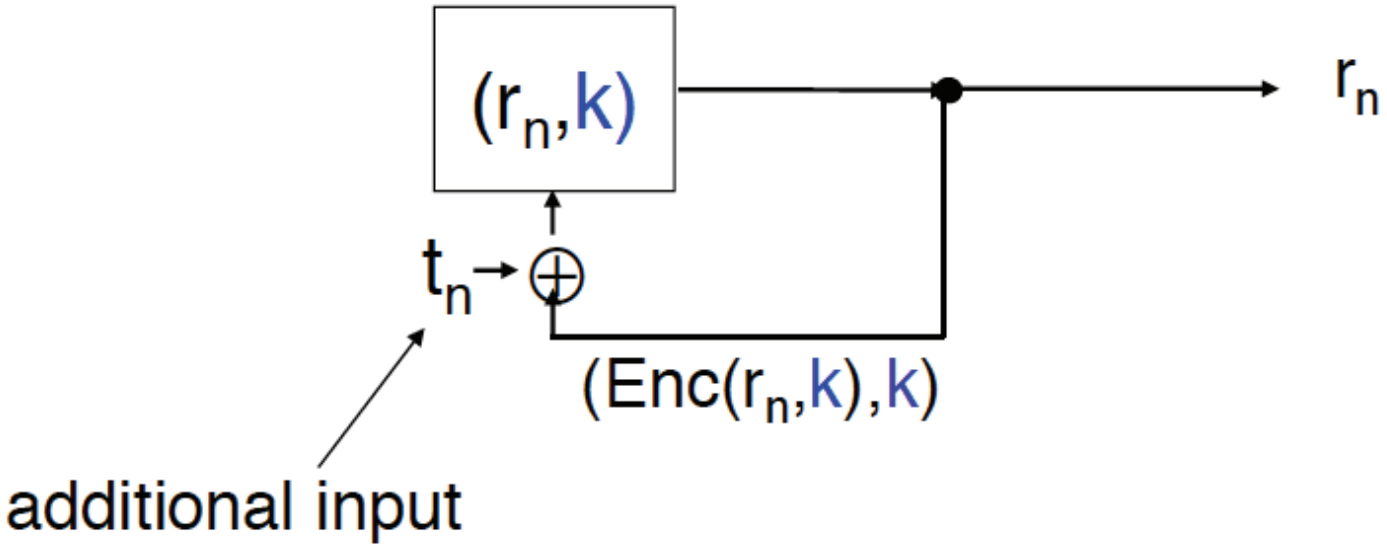
- Additional input may be provided
  - After each step
  - Occasionally
  - Upon external request of an application

Hybrid DRNGs

- Additional input may be provided
  - After each step
  - Occasionally
  - Upon external request of an application
- Additional input may have
  - Large entropy per bit, using a strong physical RNG
  - Low entropy, time etc

Block Cipher Hybrid DRNGs

- Strong block cipher, e.g., AES or Triple-DES
- Key k is kept secret



- The algorithmic part guarantees R1 and R2
- Additional input large entropy may ensure R3 and R4

Elliptic Curve DRNGs

- There are three existing proposals
  - Linear congruential generator
  - Power generator
  - Naor-Reingold generator

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

- 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
- Our new work
  - New pure and hybrid DRNG over Edward Curves
  - New complexity results

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:

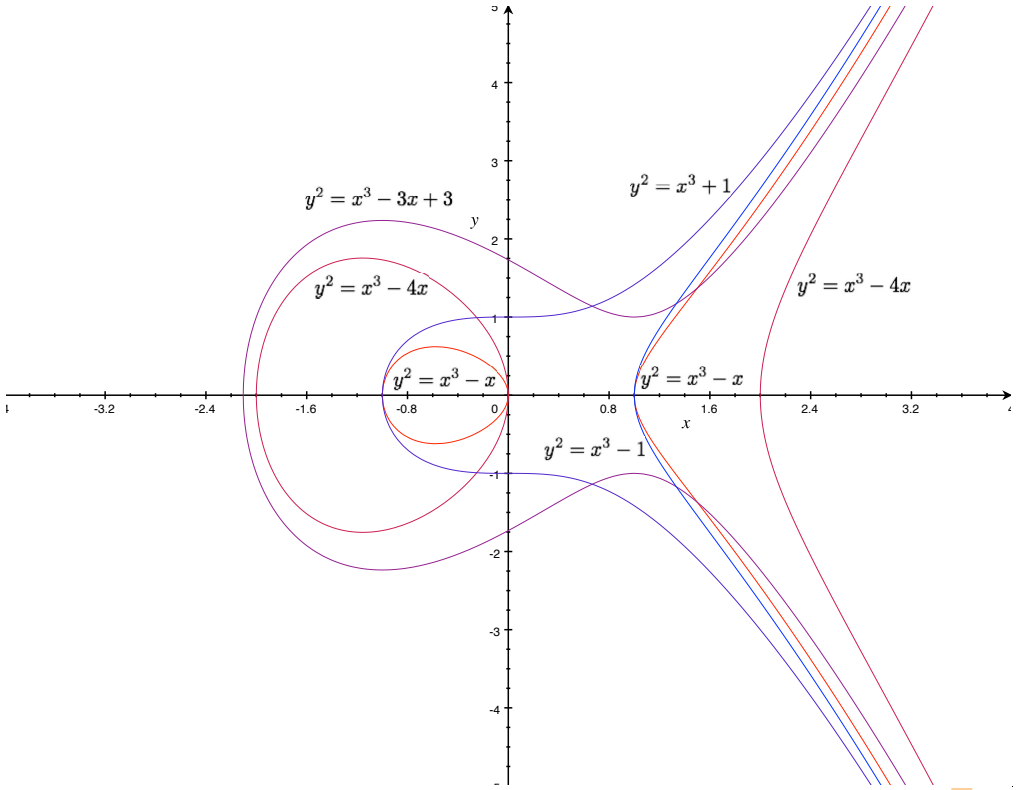
$$\boxed{\text{Find } d, \text{ given } P \text{ and } Q = [d]P}$$

Weierstrass Elliptic Curves

- The set of points (x, y) on elliptic curve together with the point at infinity $\mathcal{O}$

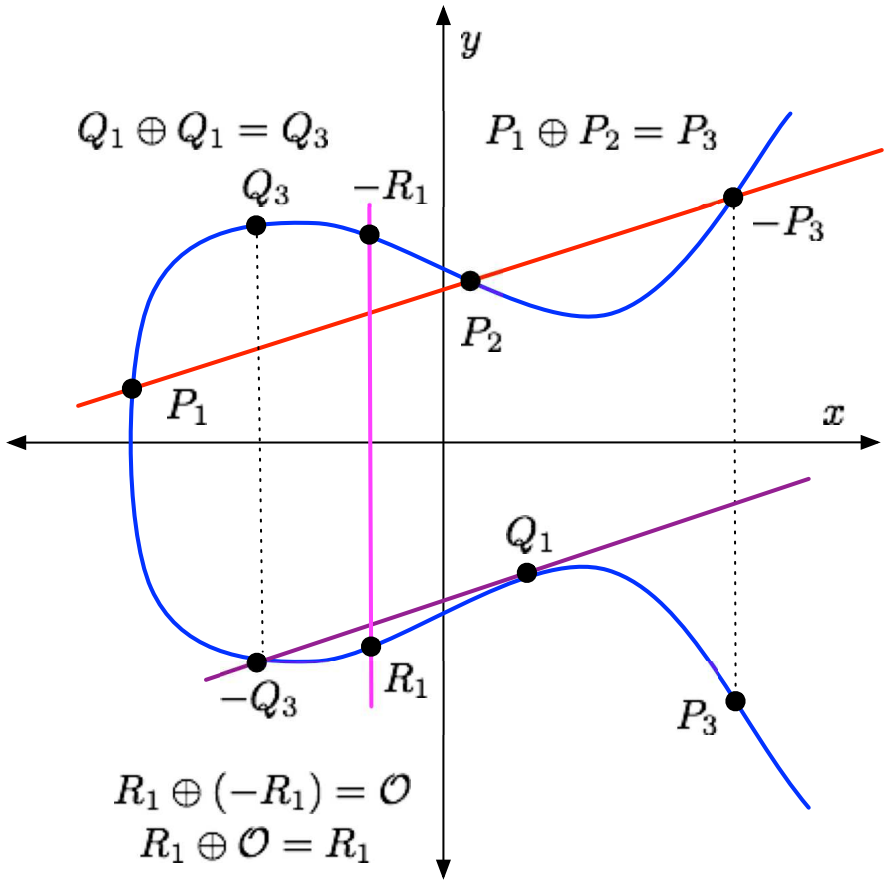
$$\mathcal{E} = \{(x, y) \mid (x, y) \in \mathcal{F}_p^2 \text{ and } y^2 = x^3 + ax + b\} \cup \{\mathcal{O}\}$$

forms an Abelian group with respect to the addition operation $\oplus$



Elliptic Curve Point Addition

- The addition operation computes the coordinates (x_3, y_3) of P_3 for $P_3 = P_1 \oplus P_2 = (x_1, y_1) \oplus (x_2, y_2)$



Elliptic Curve Addition and Doubling over $\mathcal{F}_p$

Given $P_1 = (x_1, y_1)$ and $P_2 = (x_2, y_2)$, the computation of $P_3 = (x_3, y_3)$:

- If $(x_1, y_1) = \mathcal{O}$, then $(x_3, y_3) = (x_2, y_2)$ since $P_3 = \mathcal{O} + P_2 = P_2$
- If $(x_2, y_2) = \mathcal{O}$, then $(x_3, y_3) = (x_1, y_1)$ since $P_3 = P_1 + \mathcal{O} = P_1$
- If $x_2 = x_1$ and $y_2 = -y_1$, then $(x_3, y_3) = \mathcal{O}$ since $P_3 = -P_1 + P_1 = \mathcal{O}$
- Otherwise, first compute the slope using

$$m = \begin{cases} \frac{y_2 - y_1}{x_2 - x_1} & \text{for } x_1 \neq x_2 \\ \frac{3x_1^2 + a}{2y_1} & \text{for } x_1 = x_2 \text{ and } y_1 = y_2 \end{cases}$$

- Then, (x_3, y_3) is computed using

$$\begin{aligned} x_3 &= m^2 - x_1 - x_2 \\ y_3 &= m\,(x_1 - x_3) - y_1 \end{aligned}$$

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

- 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
- 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

Edwards Curves

- Harold Edwards introduced a new normal form for elliptic curves and gave an addition law which is remarkably symmetric and much simpler
- The original form the equation Edwards studied was

$$x^2 + y^2 = c^2 + c^2 x^2 y^2$$

  solved over a field F whose characteristic is not equal to 2
- Studies on such groups go as far back as to Gauss
- Bernstein and Lange gave a slightly simpler form

$$x^2 + y^2 = 1 + dx^2 y^2$$

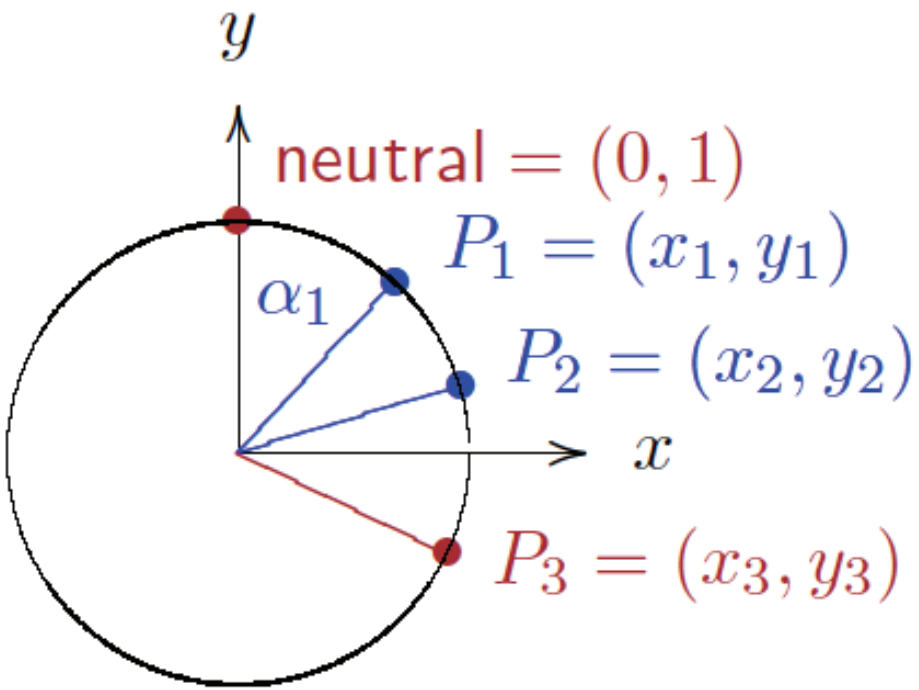
  where d is a quadratic non residue

Edwards Curves ($d = 0$)

- When $d = 0$, this becomes the unit circle
- The zero element of the group is $(0, 1)$
- The addition law is given as

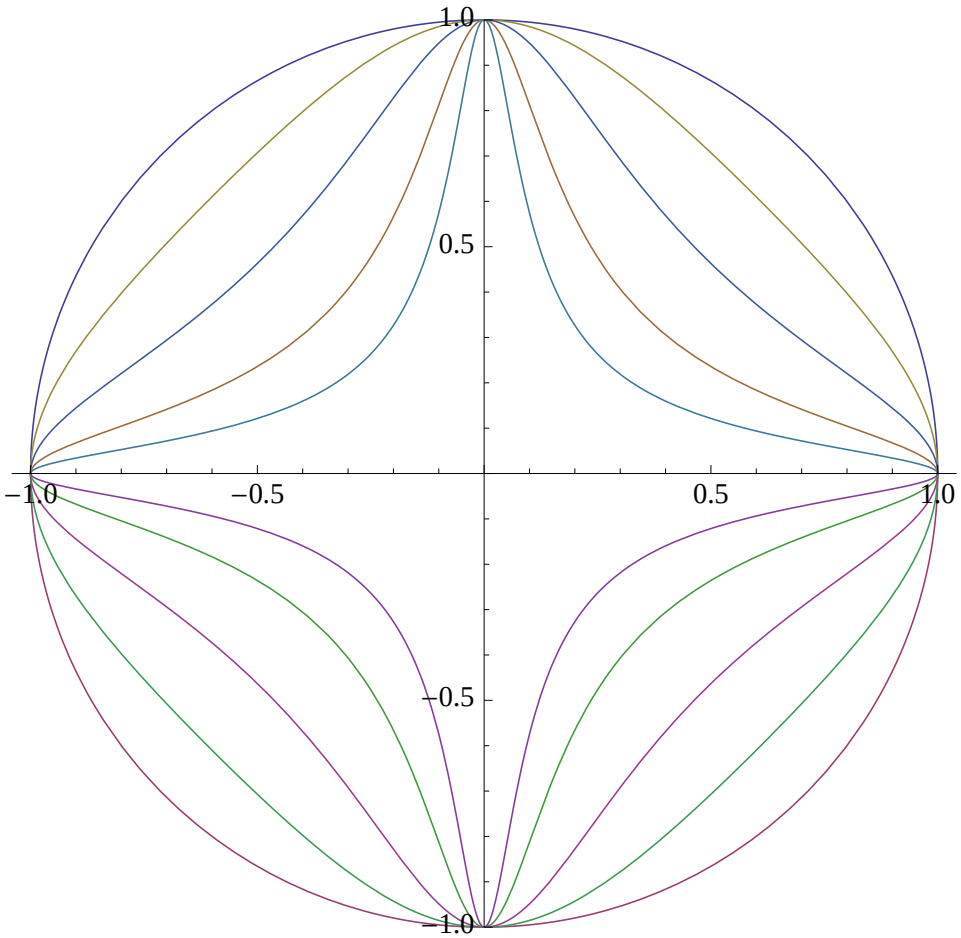
$$(x_1, y_1) \oplus (x_2, y_2) = (x_1 y_2 + x_2 y_1, y_1 y_2 - x_1 x_2)$$

- The geometric interpretation: add the angles of the points P_1 and P_2



Edwards Curves

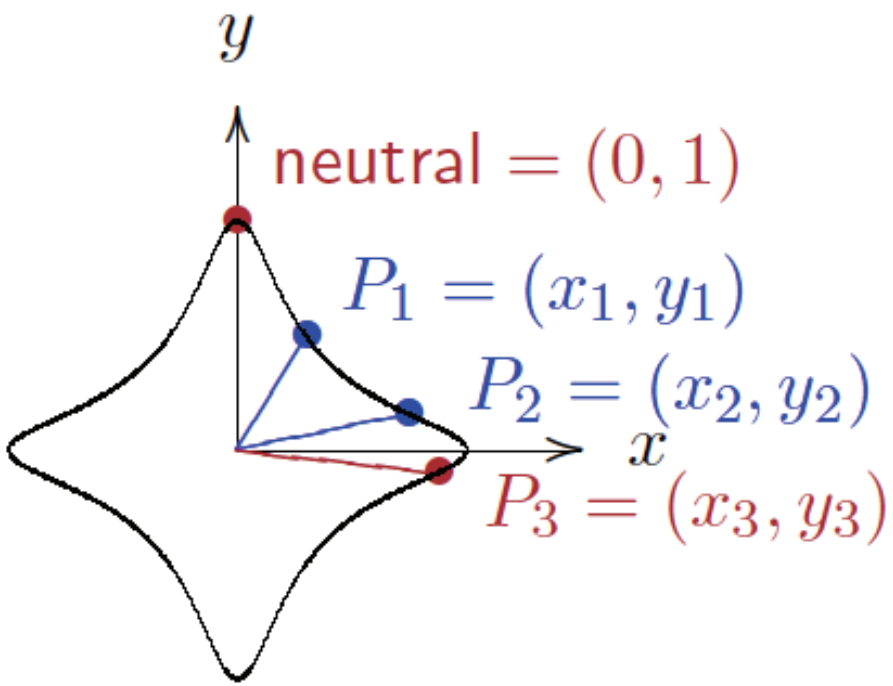
- Other values of $d \in F - \{0, 1\}$ for a non-binary field $\mathcal{F}$ form curves within the unit circle
- Edwards curves for $d = 0, -2, -10, -50, -200$



Edwards Curve Addition Law

- The zero (neutral) element is $(0, 1)$
- The inverse of (x, y) is $(-x, y)$
- The addition law

$$(x_1, y_1) \oplus (x_2, y_2) = \left(\frac{x_1 y_2 + x_2 y_1}{1 + d x_1 x_2 y_1 y_2} \ , \ \frac{y_1 y_2 - x_1 x_2}{1 - d x_1 x_2 y_1 y_2} \right)$$



Edwards Curve Projections

- A point (x, y) on the Edwards curve E_d projects to the point (u, v) in the same quadrant on the unit circle as $(u, v) = (\alpha x, \alpha y)$, where

$$\alpha = \frac{1}{\sqrt{x^2 + y^2}}$$

Edwards Curve Projections

- A point (x, y) on the Edwards curve E_d projects to the point (u, v) in the same quadrant on the unit circle as $(u, v) = (\alpha x, \alpha y)$, where

$$\alpha = \frac{1}{\sqrt{x^2 + y^2}}$$

- A point (u, v) on the unit circle projects back to the point (x, y) in the same quadrant on the Edwards curve E_d as $(x, y) = (\beta u, \beta v)$, where

$$\beta = \frac{\sqrt{2}}{\sqrt{1 + \sqrt{1 - 4du^2v^2}}}$$

Edwards Curve Projections

- A point (x_0, y_0) on the Edwards curve E_{d_0} projects to the point (x_1, y_1) in the same quadrant on the Edwards curve E_{d_1} as $(x_1, y_1) = (\gamma x_0, \gamma x_1)$, where

$$\gamma = \frac{\sqrt{2}}{\sqrt{x_0^2 + y_0^2 + \sqrt{(x_0^2 + y_0^2)^2 - 4d_1 x_0^2 y_0^2}}}$$

Edwards Curve Projections

- A point (x_0, y_0) on the Edwards curve E_{d_0} projects to the point (x_1, y_1) in the same quadrant on the Edwards curve E_{d_1} as $(x_1, y_1) = (\gamma x_0, \gamma x_1)$, where

$$\gamma = \frac{\sqrt{2}}{\sqrt{x_0^2 + y_0^2 + \sqrt{(x_0^2 + y_0^2)^2 - 4d_1 x_0^2 y_0^2}}}$$

- Equations are more complicated however where it is possible to project from one curve to another, using clock addition and roots of unity

$$\left(\sin\left(\frac{2\pi}{n}\right), \cos\left(\frac{2\pi}{n}\right)\right)$$

Edwards Curve Hybrid DRNG

- Given the seed vector $K = (k_0, k_1, \ldots, k_n)$, consider the Edwards curve E_d over the odd prime p

Edwards Curve Hybrid DRNG

- Given the seed vector $K = (k_0, k_1, \ldots, k_n)$, consider the Edwards curve E_d over the odd prime p
- Take initial $d_0 = d$ and start with $P_0 = (x_0, y_0)$ on E_{d_0}, and compute

$$[k_0]P_0 = (x_0, y_0) \oplus_{d_0} \cdots \oplus_{d_0} (x_0, y_0) = (a_0, b_0)$$

Edwards Curve Hybrid DRNG

- Given the seed vector $K = (k_0, k_1, \ldots, k_n)$, consider the Edwards curve E_d over the odd prime p
- Take initial $d_0 = d$ and start with $P_0 = (x_0, y_0)$ on E_{d_0}, and compute

$$[k_0]P_0 = (x_0, y_0) \oplus_{d_0} \cdots \oplus_{d_0} (x_0, y_0) = (a_0, b_0)$$

- Obtain t_0 from the entropy source such that $a_0^2 + b_0^2 - t_0^2$ is a QNR

Edwards Curve Hybrid DRNG

- Given the seed vector $K = (k_0, k_1, \ldots, k_n)$, consider the Edwards curve E_d over the odd prime p
- Take initial $d_0 = d$ and start with $P_0 = (x_0, y_0)$ on E_{d_0}, and compute

$$[k_0]P_0 = (x_0, y_0) \oplus_{d_0} \cdots \oplus_{d_0} (x_0, y_0) = (a_0, b_0)$$

- Obtain t_0 from the entropy source such that $a_0^2 + b_0^2 - t_0^2$ is a QNR
- Assuming $a_0 b_0 \neq 0$, set

$$d_1 = \frac{t_0^2}{a_0^2 b_0^2}(a_0^2 + b_0^2 - t_0^2)$$

Edwards Curve Hybrid DRNG

- This d_1 is a QNR, and we can use it to define a new Edwards curve

Edwards Curve Hybrid DRNG

- This d_1 is a QNR, and we can use it to define a new Edwards curve
- The selection $t_0 = 1$ always works, but this gives $d_1 = d_0$

Edwards Curve Hybrid DRNG

- This d_1 is a QNR, and we can use it to define a new Edwards curve
- The selection $t_0 = 1$ always works, but this gives $d_1 = d_0$
- Now, project the point (a_0, b_0) on the Edwards curve E_{d_1} by

$$P_1 = (x_1, y_1) = (\gamma_0 a_0, \gamma_0 b_0)$$

Edwards Curve Hybrid DRNG

- This d_1 is a QNR, and we can use it to define a new Edwards curve
- The selection $t_0 = 1$ always works, but this gives $d_1 = d_0$
- Now, project the point (a_0, b_0) on the Edwards curve E_{d_1} by

$$P_1 = (x_1, y_1) = (\gamma_0 a_0, \gamma_0 b_0)$$

- The projection coefficient simplifies $\gamma_0 = t_0^{-1}$
- Now compute

$$[k_1]P_1 = (x_1, y_1) \oplus_{d_1} \cdots \oplus_{d_1} (x_1, y_1) = (a_1, b_1)$$

Edwards Curve Hybrid DRNG

- Assuming $a_1 b_1 \neq 0$, the parameter for the next Edwards curve is

$$d_2 = \frac{t_1^2}{a_1^2 b_1^2}(a_1^2 + b_1^2 - t_1^2)$$

such that $a_1^2 + b_1^2 - t_1^2$ is a QNR and t_1 comes from the entropy source

Edwards Curve Hybrid DRNG

- Assuming $a_1 b_1 \neq 0$, the parameter for the next Edwards curve is

$$d_2 = \frac{t_1^2}{a_1^2 b_1^2}(a_1^2 + b_1^2 - t_1^2)$$

such that $a_1^2 + b_1^2 - t_1^2$ is a QNR and t_1 comes from the entropy source

- Now compute $[k_2]P_2$ on E_{d_2} where P_2 is the projection

$$P_2 = (x_2, y_2) = (t_1^{-1} a_1, t_1^{-1} b_1)$$

and so on

Edwards Curve Hybrid DRNG

- Assuming $a_1 b_1 \neq 0$, the parameter for the next Edwards curve is

$$d_2 = \frac{t_1^2}{a_1^2 b_1^2}(a_1^2 + b_1^2 - t_1^2)$$

  such that $a_1^2 + b_1^2 - t_1^2$ is a QNR and t_1 comes from the entropy source
- Now compute $[k_2]P_2$ on E_{d_2} where P_2 is the projection

$$P_2 = (x_2, y_2) = (t_1^{-1} a_1, t_1^{-1} b_1)$$

  and so on
- The final point produced is $[k_n]P_n$ on the Edwards curve E_{d_n}

Edwards Curve Hybrid DRNG

- System parameters: Edward curve over $\mathcal{F}_p$ with odd prime p and $d_0 = d$, and a point on the curve $P_0 = (x_0, y_0)$

Edwards Curve Hybrid DRNG

- System parameters: Edward curve over $\mathcal{F}_p$ with odd prime p and $d_0 = d$, and a point on the curve $P_0 = (x_0, y_0)$
- Input seed vector: $K = (k_0, k_1, \ldots, k_n)$

Edwards Curve Hybrid DRNG

- System parameters: Edward curve over $\mathcal{F}_p$ with odd prime p and $d_0 = d$, and a point on the curve $P_0 = (x_0, y_0)$
- Input seed vector: $K = (k_0, k_1, \ldots, k_n)$
- The entropy source produces: $T_0, T_1, \ldots, T_n$ which have been modified to obtain the QNRs $t_0, t_1, \ldots, t_n$ at each step

Edwards Curve Hybrid DRNG

- System parameters: Edward curve over $\mathcal{F}_p$ with odd prime p and $d_0 = d$, and a point on the curve $P_0 = (x_0, y_0)$
- Input seed vector: $K = (k_0, k_1, \ldots, k_n)$
- The entropy source produces: $T_0, T_1, \ldots, T_n$ which have been modified to obtain the QNRs $t_0, t_1, \ldots, t_n$ at each step
- For $i = 0$ to $i = n$, Step i:

  $(a_i, b_i) = [k_i]P_i$

  Get T_i and compute t_i such that $a_i^2 + b_i^2 - t_i^2$ is a QNR

  $d_{i+1} = \frac{t_i^2}{a_i^2 b_i^2}(a_i^2 + b_i^2 - t_i^2)$

  $\gamma_i = t_i^{-1}$

  $P_{i+1} = (\gamma_i a_i, \gamma_i b_i)$

Edwards Curve Hybrid DRNG

- System parameters: Edward curve over $\mathcal{F}_p$ with odd prime p and $d_0 = d$, and a point on the curve $P_0 = (x_0, y_0)$
- Input seed vector: $K = (k_0, k_1, \ldots, k_n)$
- The entropy source produces: $T_0, T_1, \ldots, T_n$ which have been modified to obtain the QNRs $t_0, t_1, \ldots, t_n$ at each step
- For $i = 0$ to $i = n$, Step i:

  $(a_i, b_i) = [k_i]P_i$

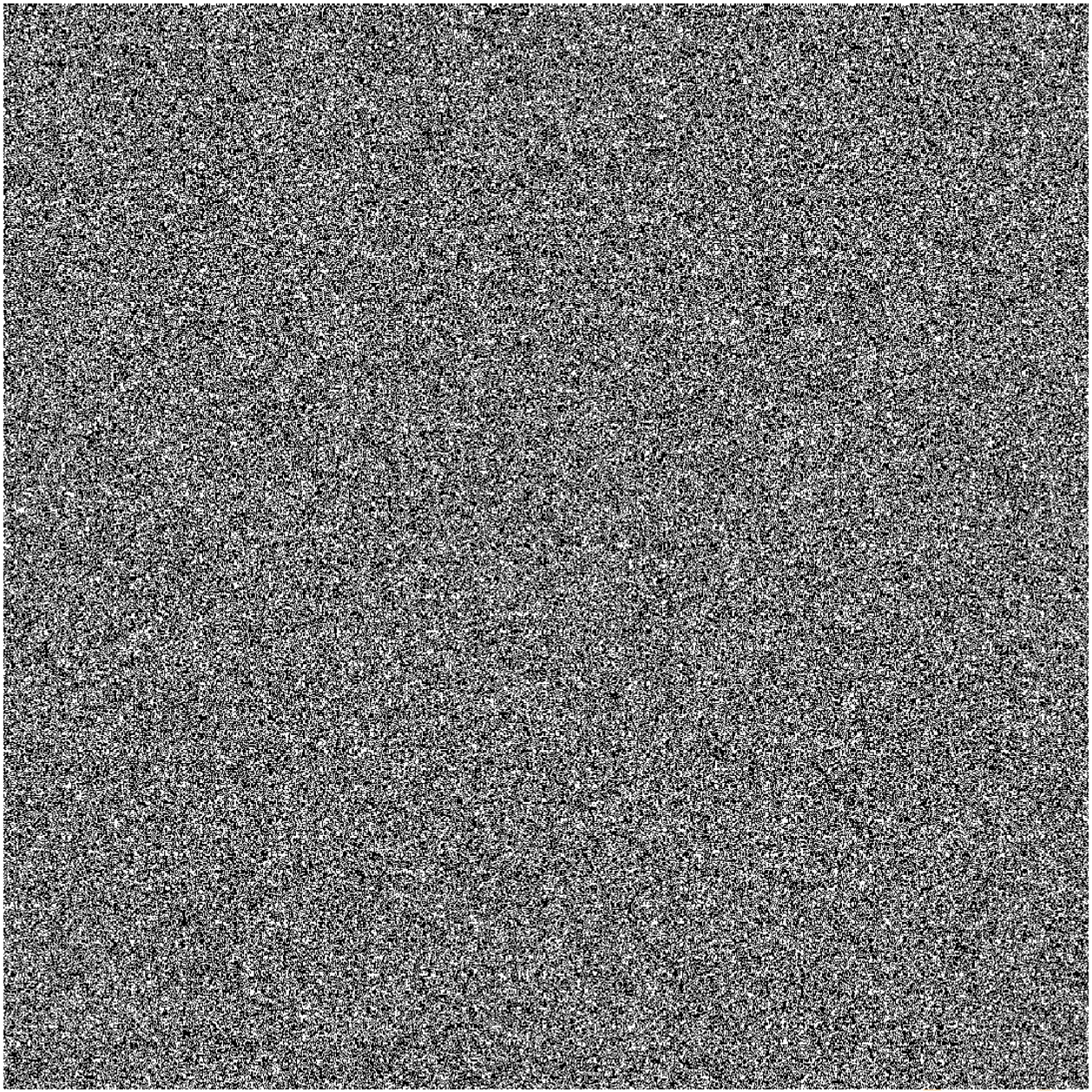
  Get T_i and compute t_i such that $a_i^2 + b_i^2 - t_i^2$ is a QNR

  $d_{i+1} = \frac{t_i^2}{a_i^2 b_i^2}(a_i^2 + b_i^2 - t_i^2)$

  $\gamma_i = t_i^{-1}$

  $P_{i+1} = (\gamma_i a_i, \gamma_i b_i)$
- Random points $P_1, P_2, \ldots, P_n$

Edwards Curve Hybrid DRNG



A Complexity Result

Theorem

Suppose p, q are prime numbers with $p = 4q - 1$. A few such (p, q) pairs are $(11, 3)$, $(19, 5)$, $(331, 83)$, $(1314883, 328721)$, $(2760727332067, 690181833017)$. Consider the Edwards group G on E_d with $d = -1$ (a QNR) over $\mathcal{F}_p$. G has identity $(0, 1)$, the element $(0, -1)$ of order 2, and the elements $(\pm 1, 0)$ are of order 4. Any other (x, y) has order q, $2q$ or $4q$.