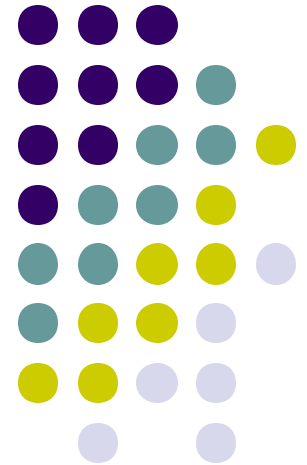


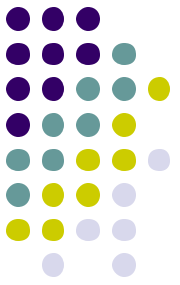
X9.82 Part 3

Number Theoretic DRBGs



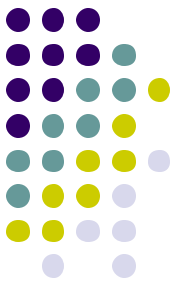
Don B. Johnson
NIST RNG Workshop
July 20, 2004





WHY?

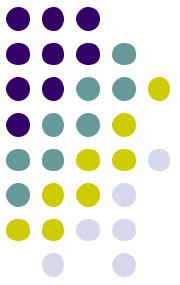
- Asymmetric key operations are about **100 times slower** than symmetric key or hash operations
- Why have 2 DRBGs based on hard problems in number theory?
- Certainly **not** expected to be chosen for performance reasons!



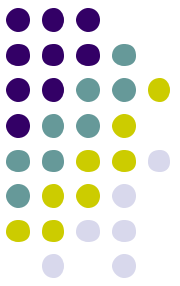
Some Possible Reasons

- 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

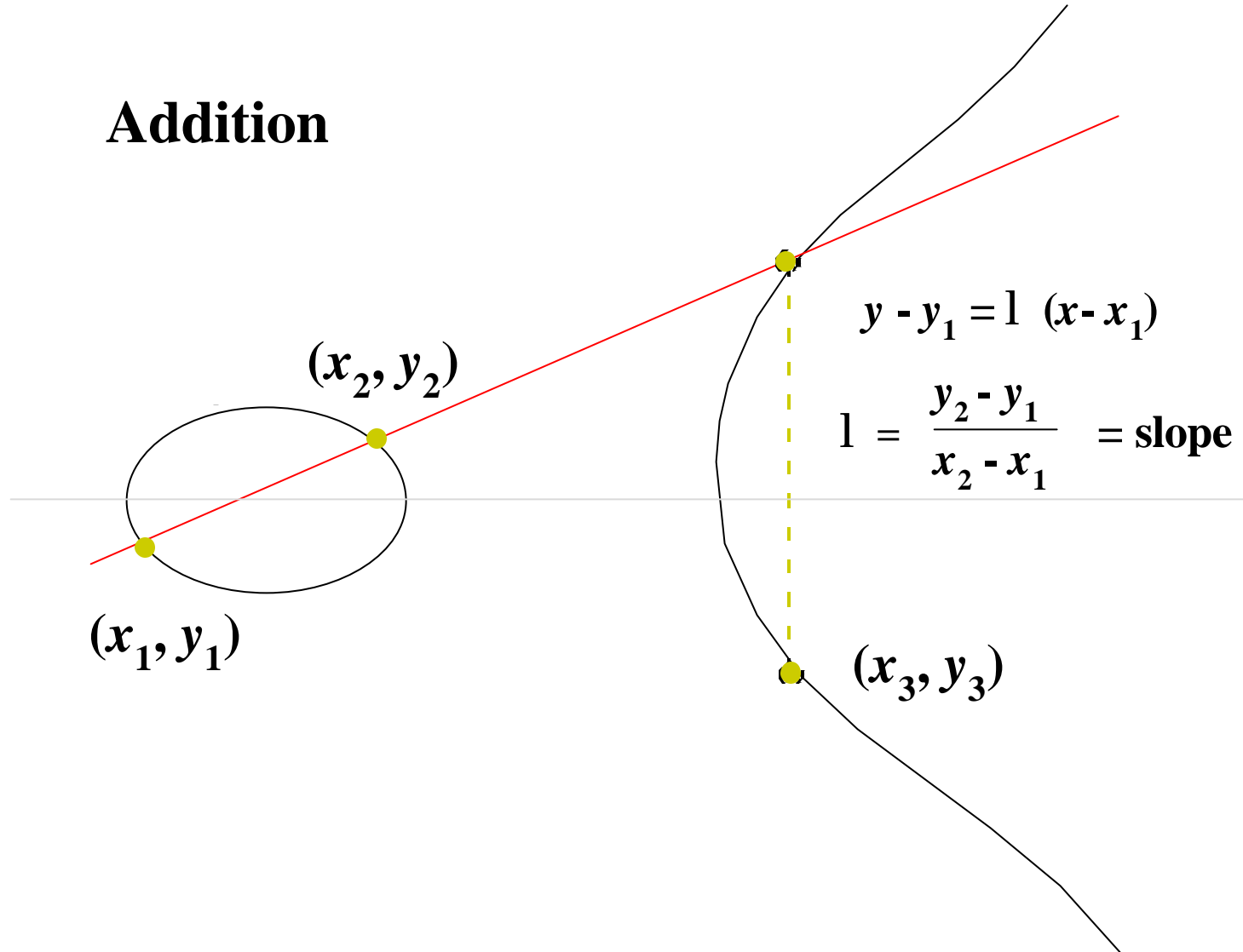


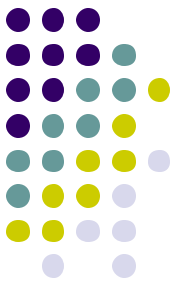
Quick Elliptic Curve Review

- 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 $y^2 = x^3 + ax + b$

Addition





Toy Example: The Field \mathbf{Z}_{23}

- The field \mathbf{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 Z_{23}^*



- Z_{23}^* consists of the 22 elements of Z_{23} excluding 0

$$5^0 = 1$$

$$5^1 = 5$$

$$5^2 = 2$$

$$5^3 = 10$$

$$5^4 = 4$$

$$5^5 = 20$$

$$5^6 = 8$$

$$5^7 = 17$$

$$5^8 = 16$$

$$5^9 = 11$$

$$5^{10} = 9$$

$$5^{11} = 22$$

$$5^{12} = 18$$

$$5^{13} = 21$$

$$5^{14} = 13$$

$$5^{15} = 19$$

$$5^{16} = 3$$

$$5^{17} = 15$$

$$5^{18} = 6$$

$$5^{19} = 7$$

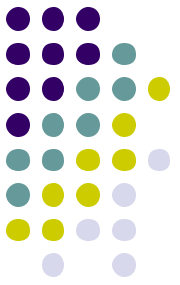
$$5^{20} = 12$$

$$5^{21} = 14$$

And return

$$5^{22} = 1$$

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

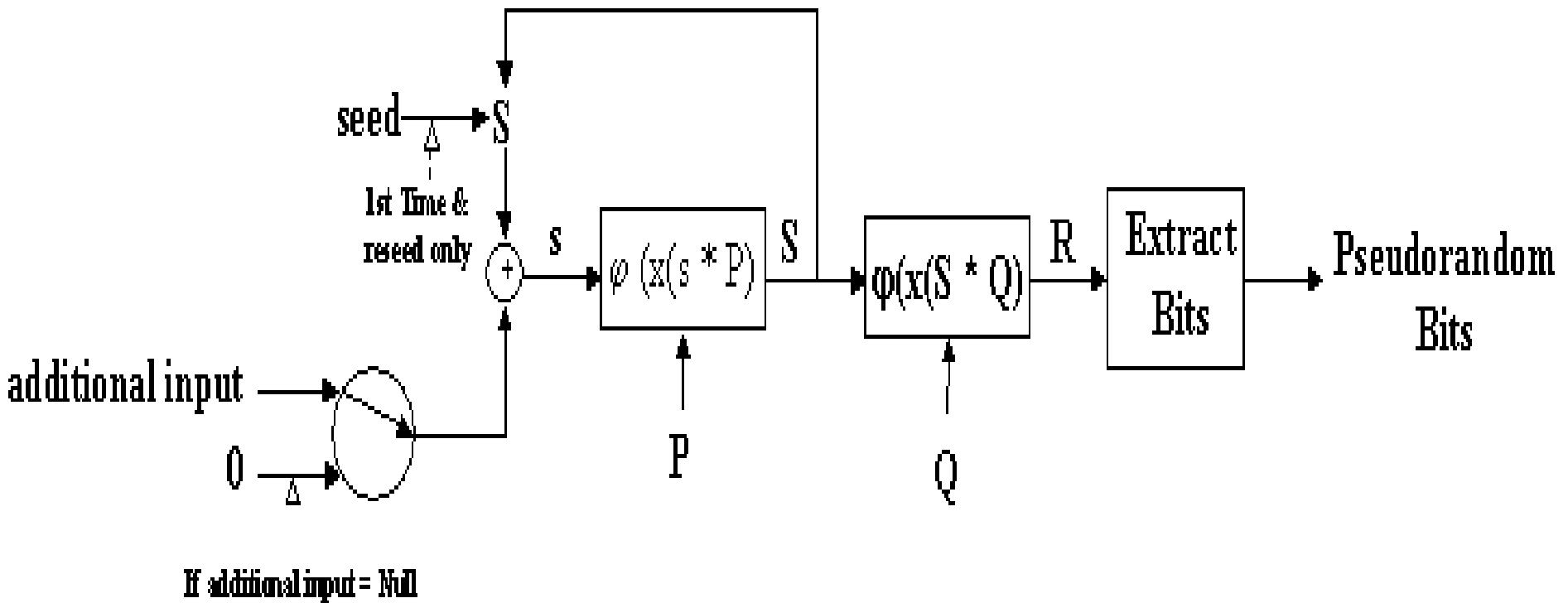
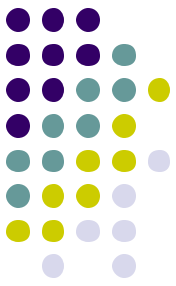


Solutions to $y^2 = x^3 + x + 1$ Over Z_{23}

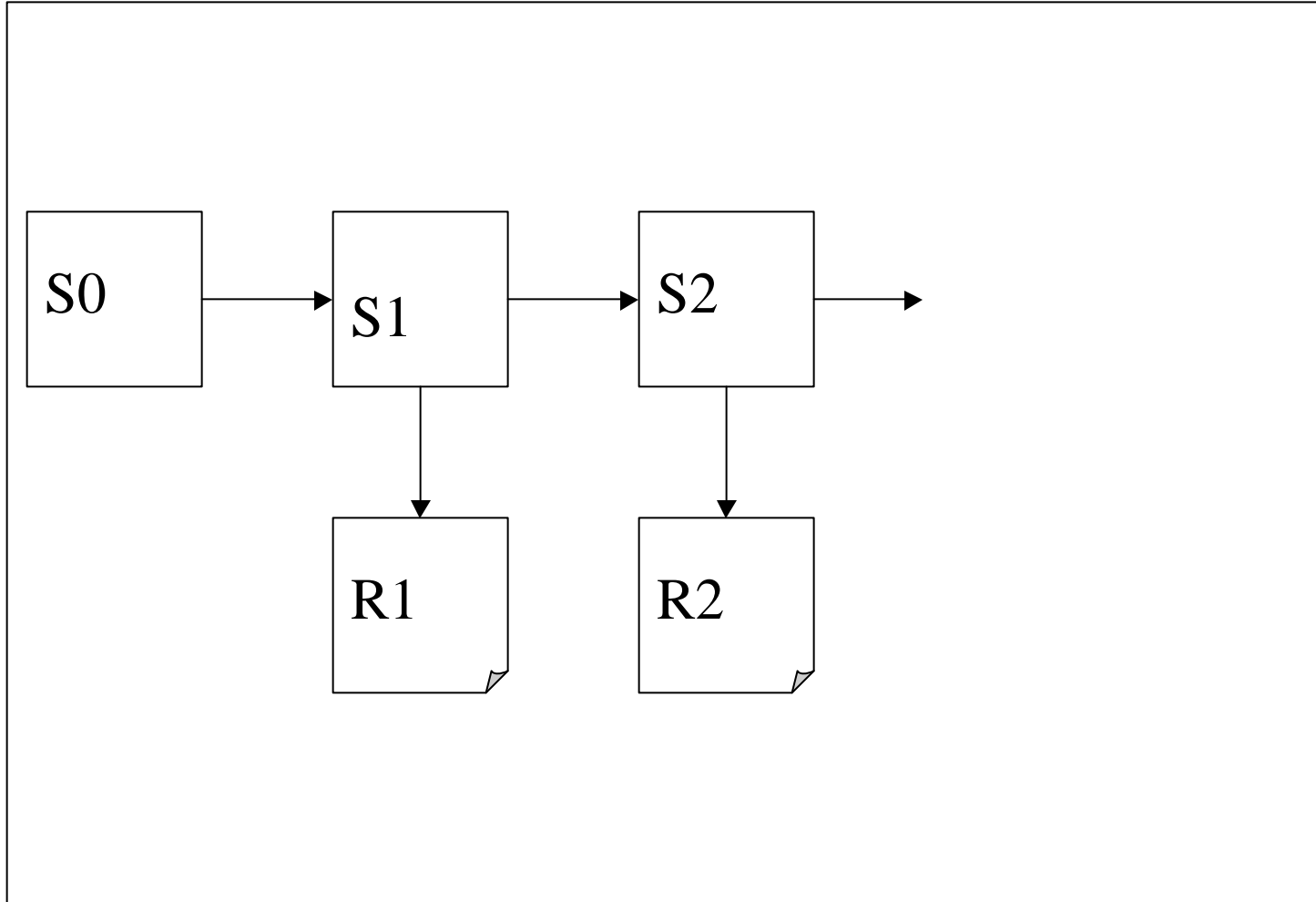
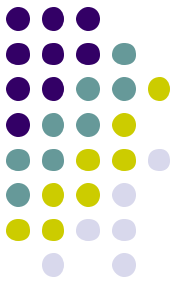
(0, 1)	(6, 4)	(12, 19)
(0, 22)	(6, 19)	(13, 7)
(1, 7)	(7, 11)	(13, 16)
(1, 16)	(7, 12)	(17, 3)
(3, 10)	(9, 7)	(17, 20)
(3, 13)	(9, 16)	(18, 3)
(4, 0)	(11, 3)	(18, 20)
(5, 4)	(11, 20)	(19, 5)
(5, 19)	(12, 4)	(19, 18)
\mathcal{A}		

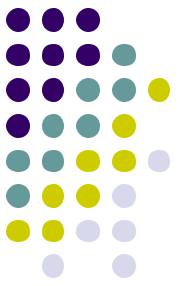
There are 28 points on this toy elliptic curve

ECC DRBG Flowchart



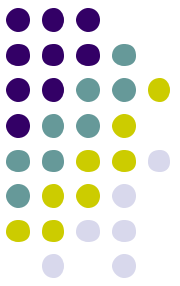
Unlooped Flowchart





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?



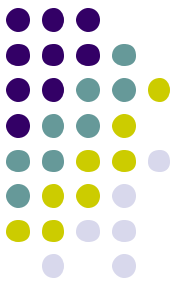
X-Coordinate Not Random

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

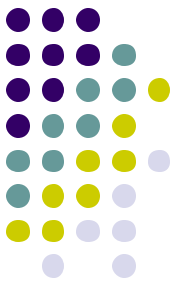
X values appearing once: 4

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

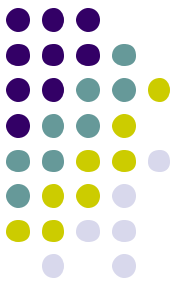
If I get first 4 bits of X value of 0100a, I know a must be a 1, as 9 exists but 8 does not



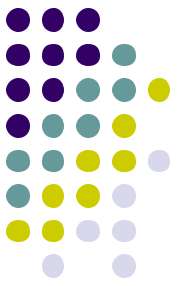
1-bit Predictability

- 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

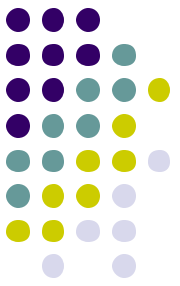


Prime field	Truncate at least 13 leftmost bits of x coordinate
Binary Field, cofactor = 2	Truncate at least 14 leftmost bits of x coordinate
Binary Field, cofactor = 4	Truncate at least 15 leftmost bits of x coordinate



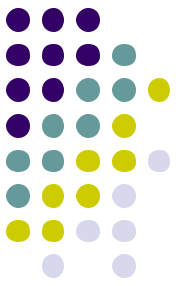
Truncation

- 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

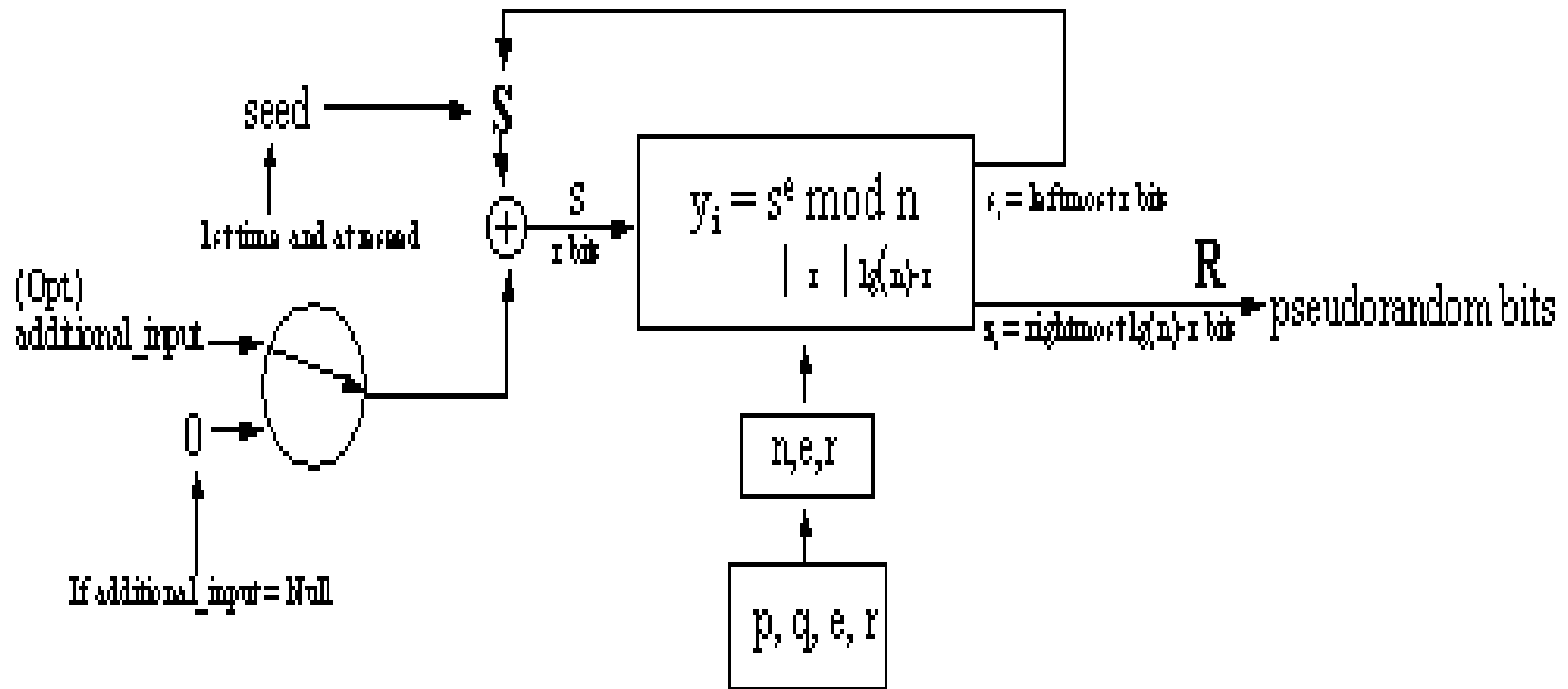


Quick RSA Review

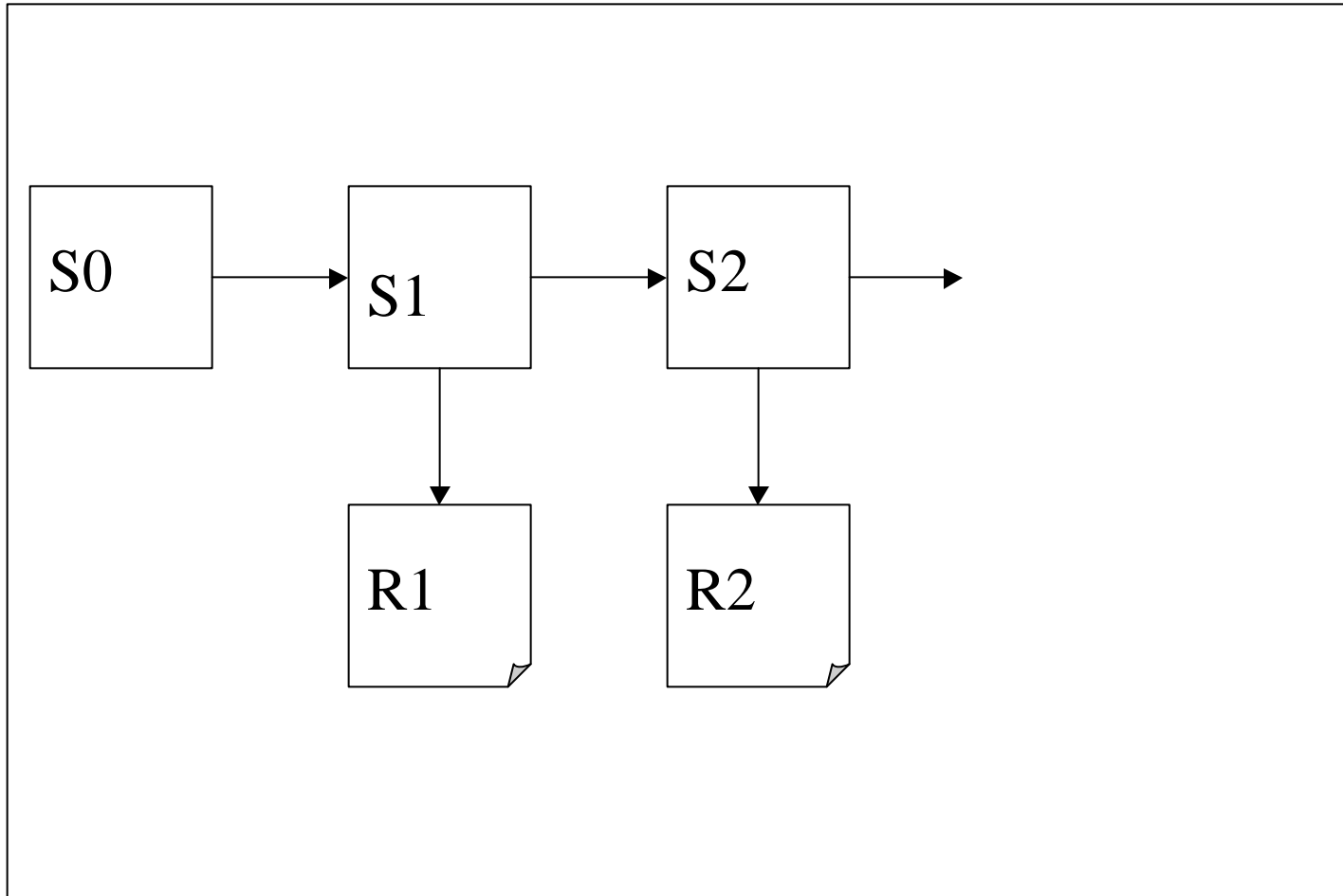
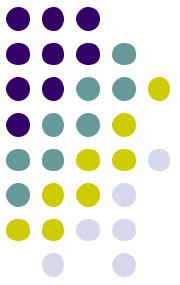
- Choose odd public exponent e and primes p and q such that e has no common factor with p or q , set $n = pq$
- Find d such $ed = 1 \pmod{(p-1)(q-1)}$
- Public key is (e, n) , private key is (d, n)
- Hard to find d from (e, n) if $n \geq 1024$ bits
- $(M^e \pmod n)$ is **hard to invert** for most M

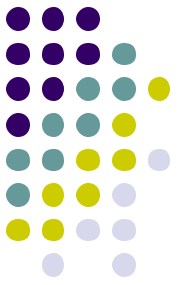


Micali-Schnorr DRBG



Unlooped Flowchart

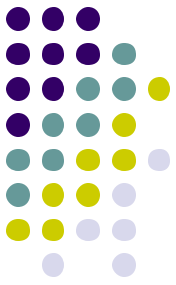




Micali-Schnorr Truncation

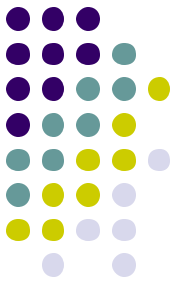
- For MS truncation, we only use the RSA **hard core bits** as random bits
- This has high assurance that the number theory problem to be solved is as hard as possible!
- Reseed after 50,000 iterations

NIST/ANSI X9 Security Levels Table



Security Levels (in bits)	ECC (order of G in bits)	MS (RSA) (modulus in bits)
80	160	1024, 10 hardcore bits
112	224	2048, 11 hardcore bits
128	256	3072, 11 hardcore bits
192	384	Not specified

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