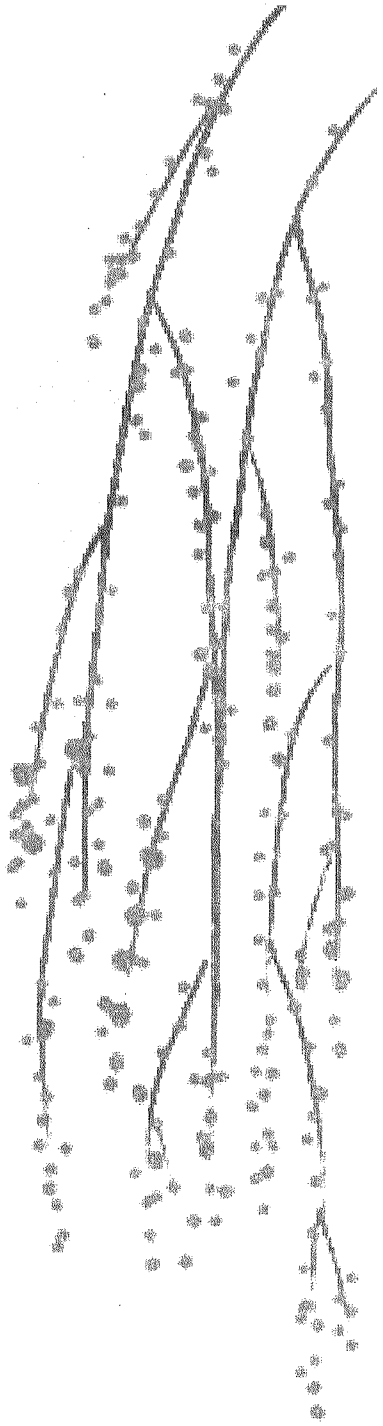
ANSI X9.82, Part 3 Deterministic Random Bit Generators (DRBGs)

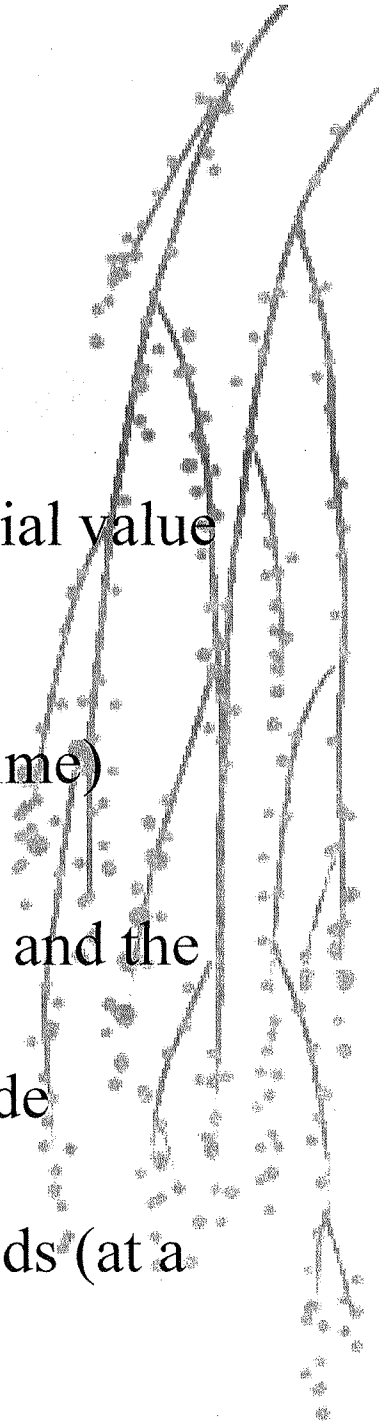
Elaine Barker, NIST

March 5-6, 2003



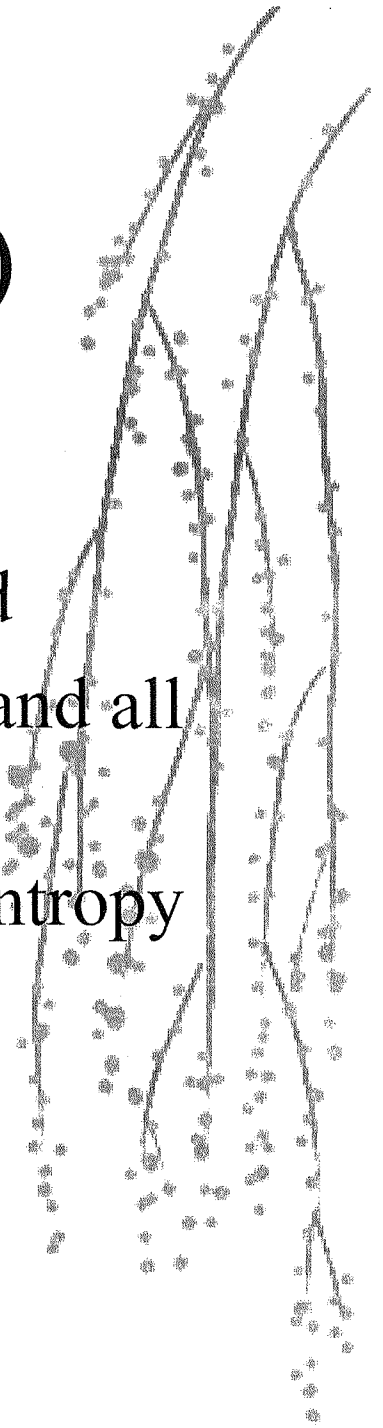
DRBG Concepts

- Input:
  - Seed or SEED SET (e.g., used to determine initial value and key(s))
  - User input
  - Time-variant information (e.g., counters, date/time)
- DRBG "instance"
  - Function of DRBG technique, crypto algorithm and the seed
  - Different instances for different purposes provide higher security – risk assessment
  - Two different instances **shall** have different seeds (at a minimum)



DRBG Concepts (Contd.)

- DRBG "state"
  - The initial state is determined by the seed
  - Subsequent states depend on the DRBG and all prior input
  - The DRBG can be reseeded – previous entropy is not "lost"



Seeds and Reseeding

- Seeds may be secret or public; an instantiation **shall not** be used to generate both
- Secret seeds **shall not** intentionally be reused.

Seeds and Reseeding (Contd.)

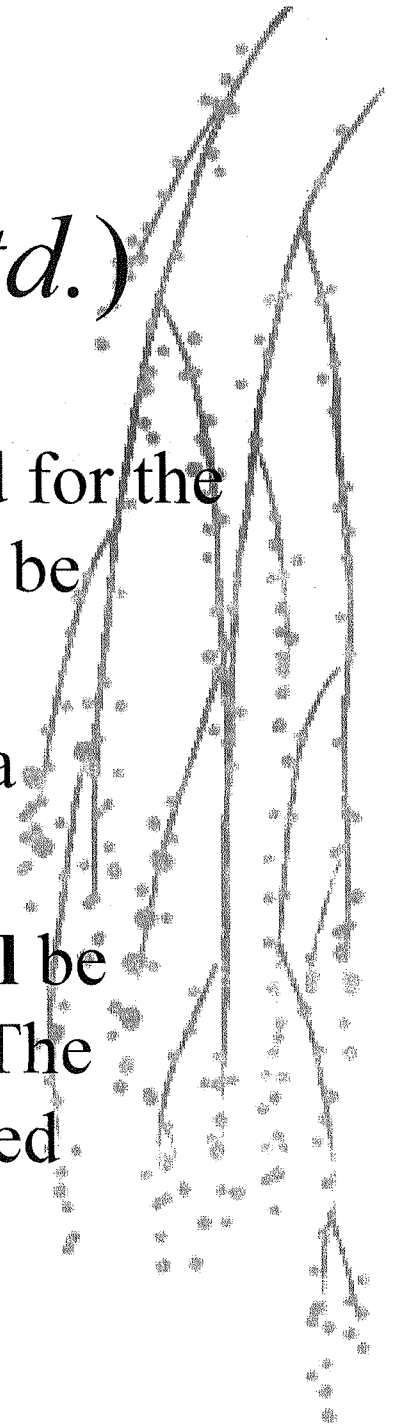
- Seed size and entropy: For x bits of security, seed size $\geq 2x$, and entropy $\geq x$.

	80	112	128	192	256
Bits of Security Strength	80	112	128	192	256
Minimum Entropy	80	112	128	192	256
Minimum Seed Size	160	224	256	320	384

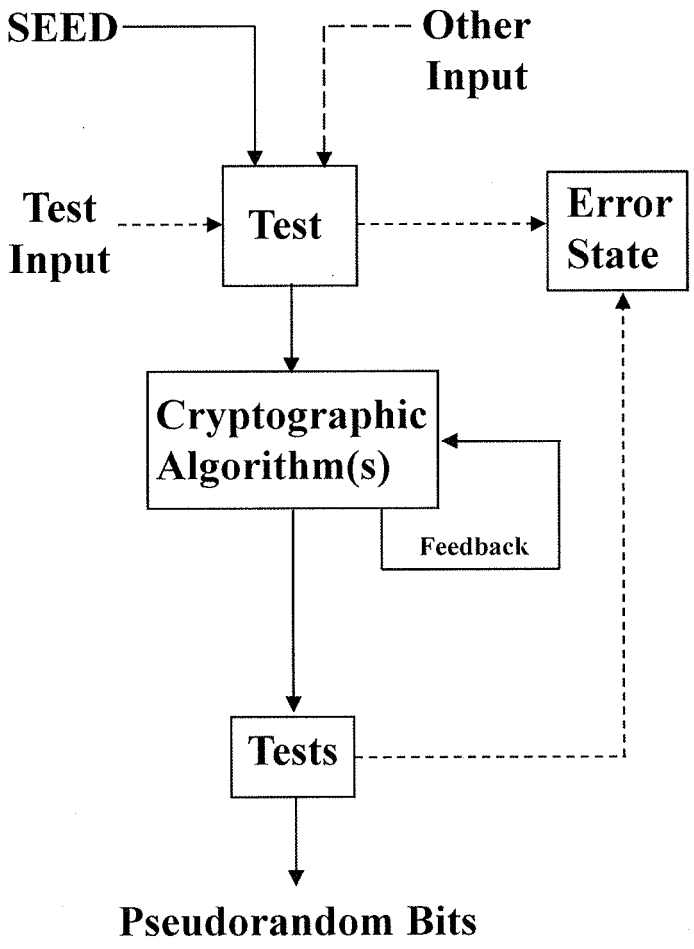
Seeds and Reseeding (Contd.)

- Seed source: Seeds **shall** be acquired from an Approved generator
- Seed privacy: Protect seeds in accordance with the security of the target data
- Reseeding: Reduces security risks. A new seeds **shall not** be identical to the old seed.

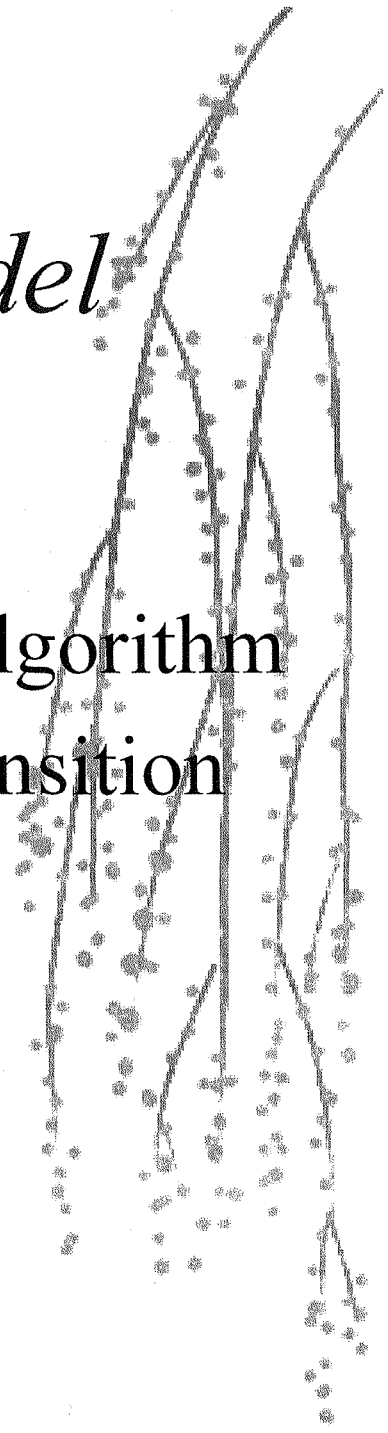
Seeds and Reseeding (Contd.)



- Seed separation: When possible, seeds used for the generation of different types of data should be different.
- Seed replacement: Secret seeds **shall** have a specified cryptoperiod.
- Keys as part of a SEED-SET: The key **shall** be independent of the rest of the SEED-SET. The SEED-SET **shall** have entropy ≥ the required strength.


SEED
Other Input
Test Input
Test
Error State
Cryptographic Algorithm(s)
Feedback
Tests
Pseudorandom Bits

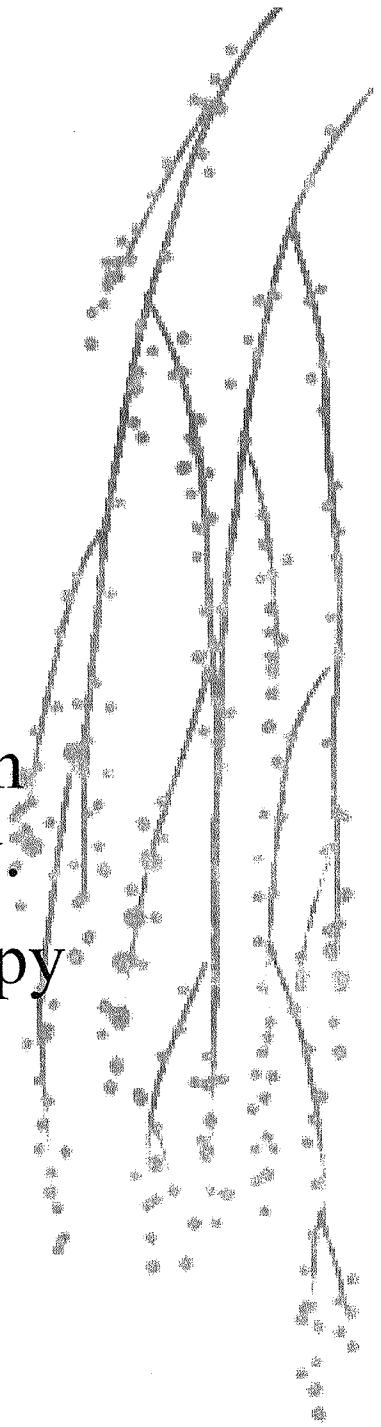
General Model



- Cryptographic algorithm
- Internal state transition
- Testing
- Error state

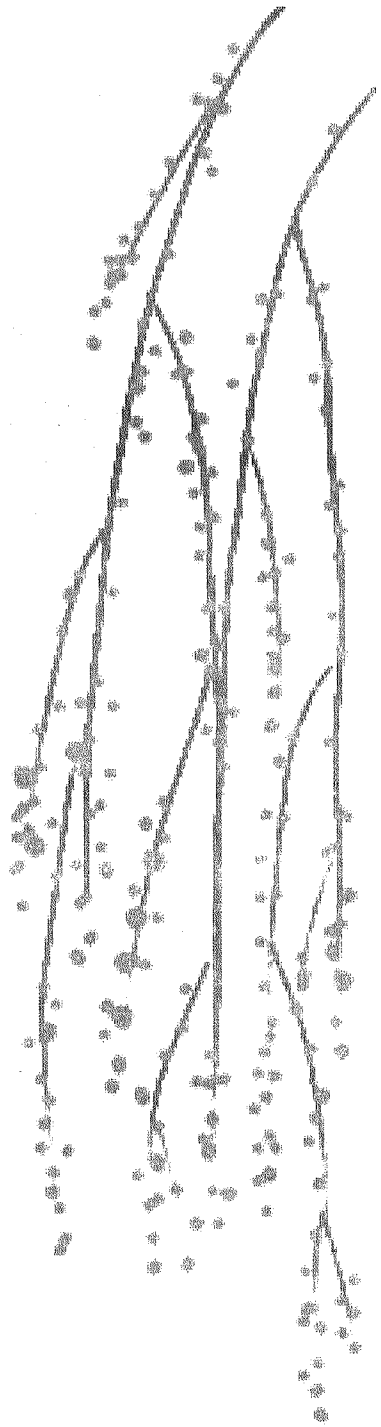
General Implementation Requirements

- When unpredictability is required, input information and internal states **shall not** be revealed.
- May transition between states on demand, in response to external events, or continuously.
- Seeds may be provided from multiple entropy sources.
- For some DRBGs, keys may be fixed.

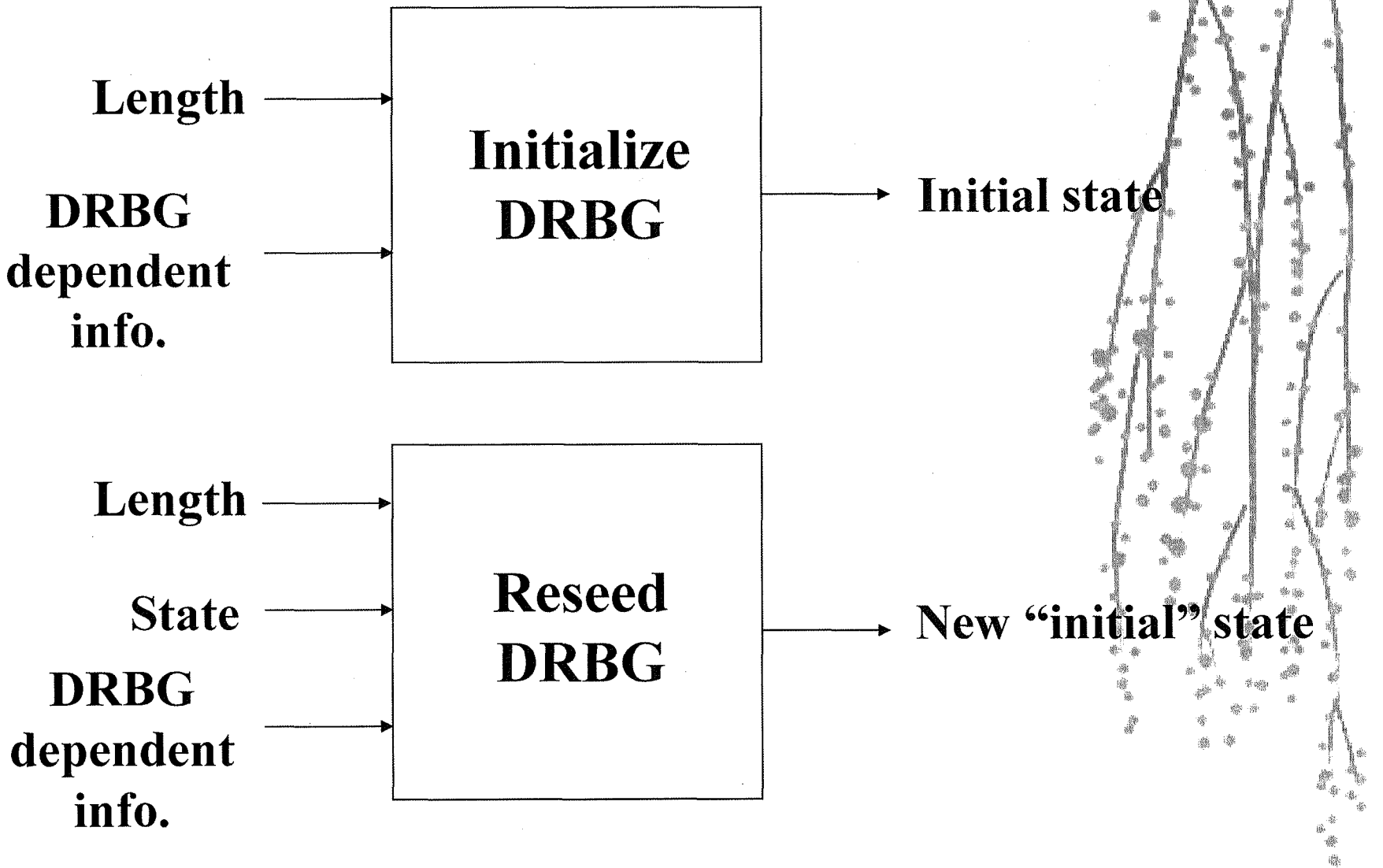


Types of DRBGs

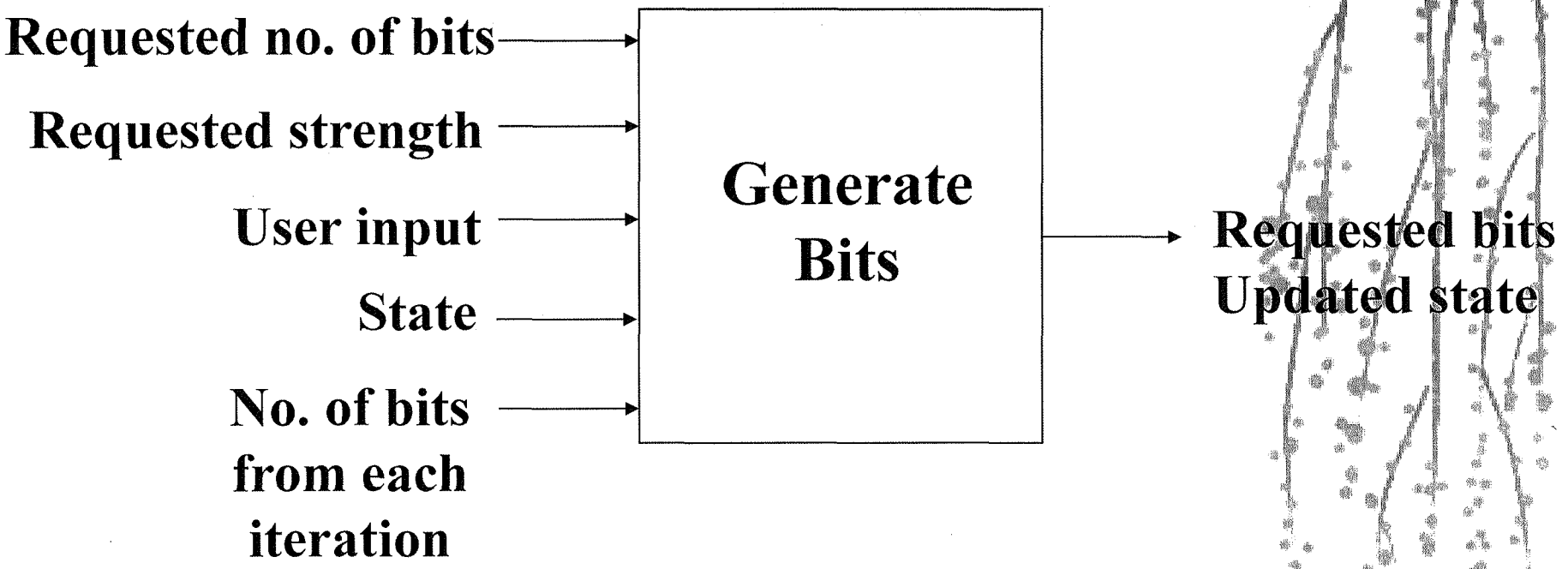
- Keyless hash DRBG
- Keyed-hash DRBG
- DRBGs based on block ciphers
- DRBGs related to number theoretic problems



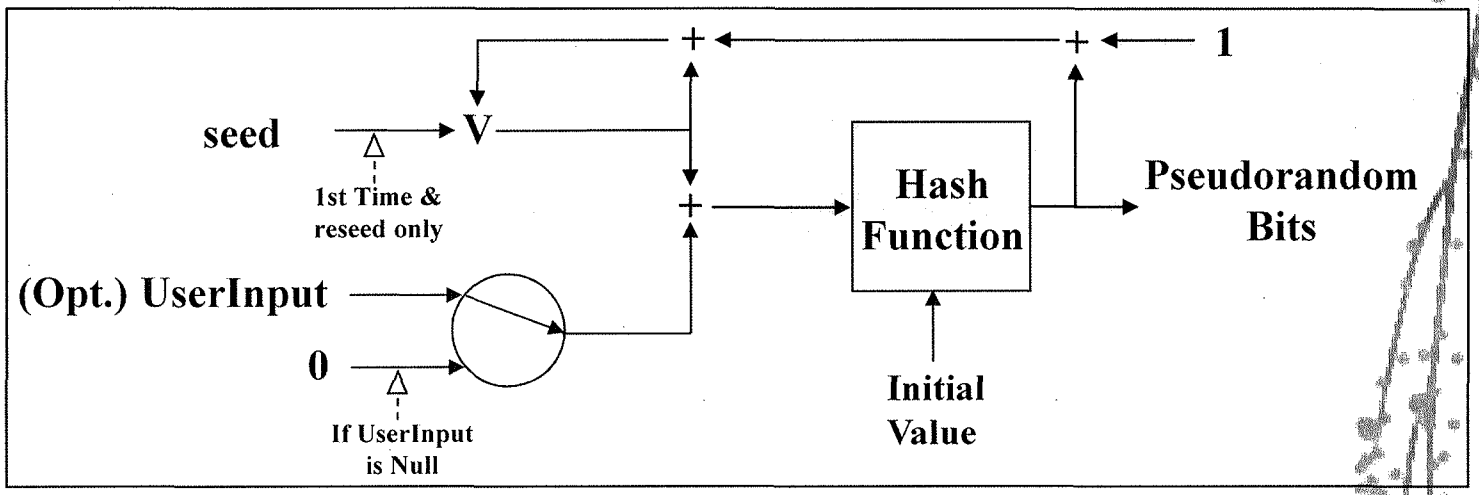
Generalized DRBG



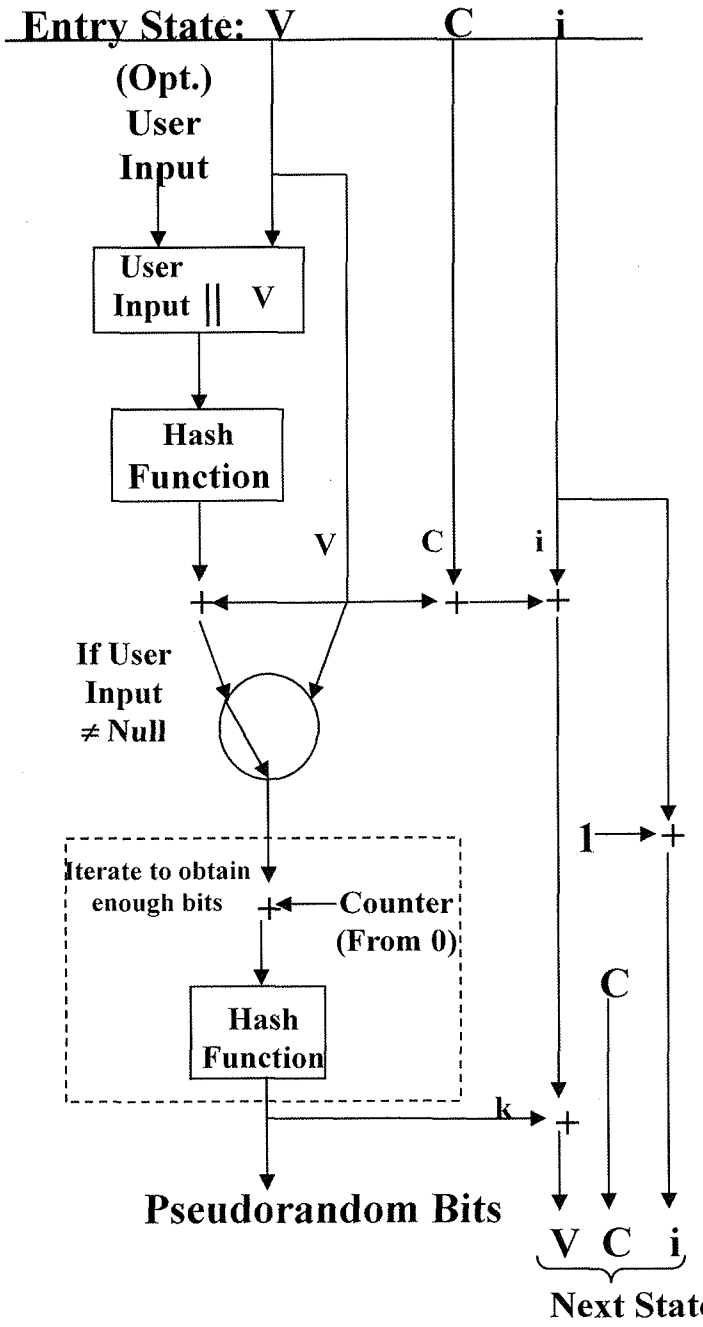
Generalized DRBG (Contd.)



SHA1KeylessHashDRBG



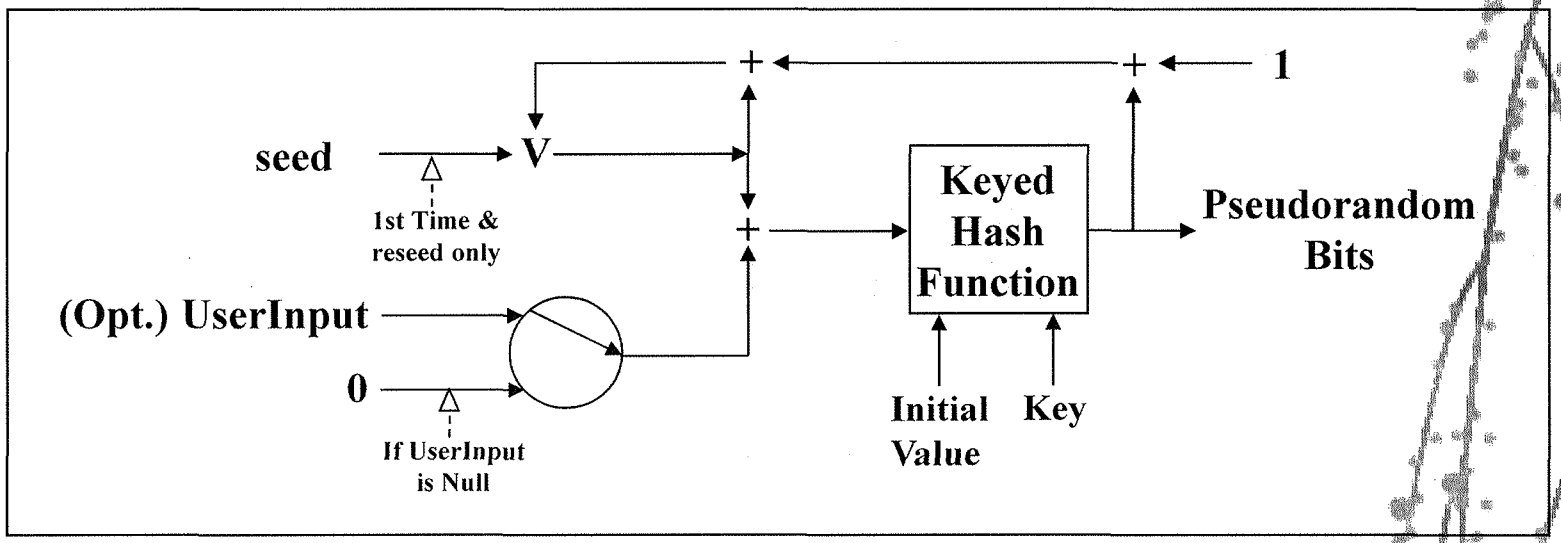
- Modified from the version contained in ANSI standards
- Seed length: $160 \leq seedlen \leq 512$ bits with at least 80 bits of entropy
- State: V
- Security strength: 80 bits



KeylessHashDRBG

- Uses any Approved hash function
- Seed length ≥ the output block size (N) with entropy $N/2$
- Choose hash function in accordance with the desired security strength
- State: V, C, i, t, *strength*
- Strength: $N/2$

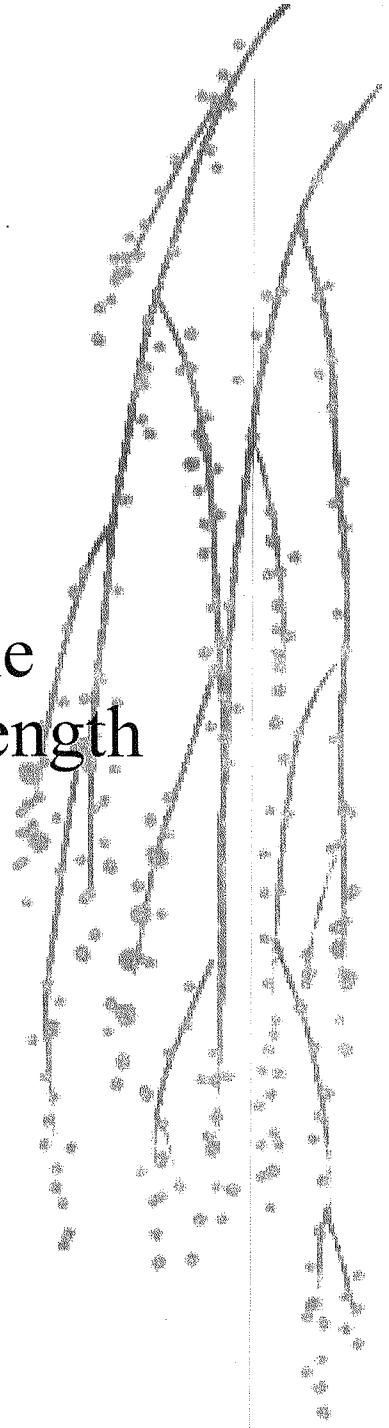
SHA1KeyedHashDRBG



- Keyed version of KeylessHashDRBG
- SEED-SET: *V*, where $160 \leq vlen \leq (512 - keylen)$ bits, and *Key*, where $160 \leq keylen \leq (512 - vlen)$ bits. Each **shall** have entropy ≥ 80 bits.
- State: *V* [, *Key*]
- Strength: 80 bits

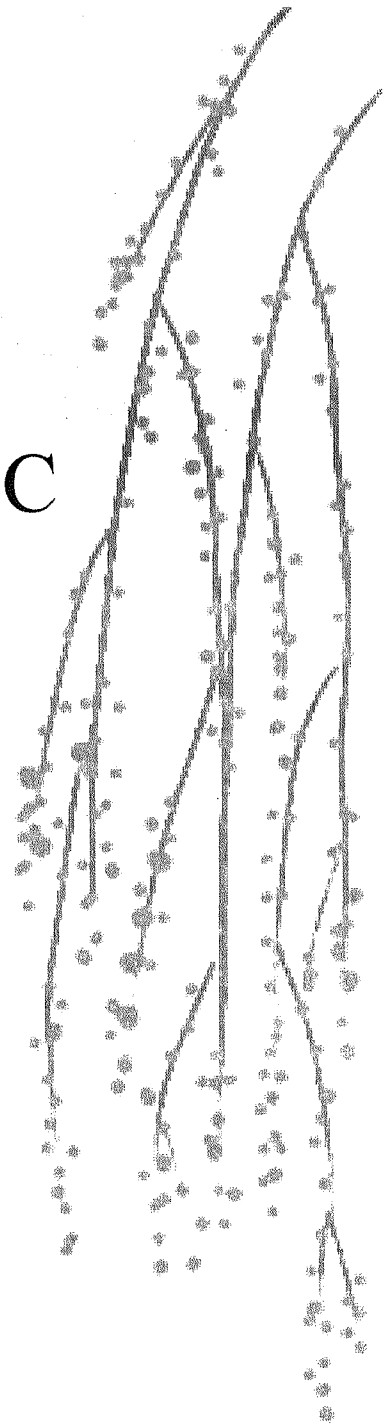
X917DRBG (Contd.)

- Seed: Used to derive *V* and 1-3 keys.
  - TDEA: *vlen* = 64, *keylen* = 112 or 168
  - AES: *vlen* = 128, *keylen* = 128, 192 or 256
- Seed length and entropy: *seedlen* ≥ twice the security strength, entropy ≥ the security strength
  - 2TDEA: *seedlen* ≥ 160, entropy ≥ 80
  - 3TDEA: *seedlen* ≥ 224, entropy ≥ 112
  - AES-128: *seedlen* ≥ 256, entropy ≥ 128
  - AES-192: *seedlen* ≥ 384, entropy ≥ 192
  - *AES-256: seedlen* ≥ 512, entropy ≥ 256

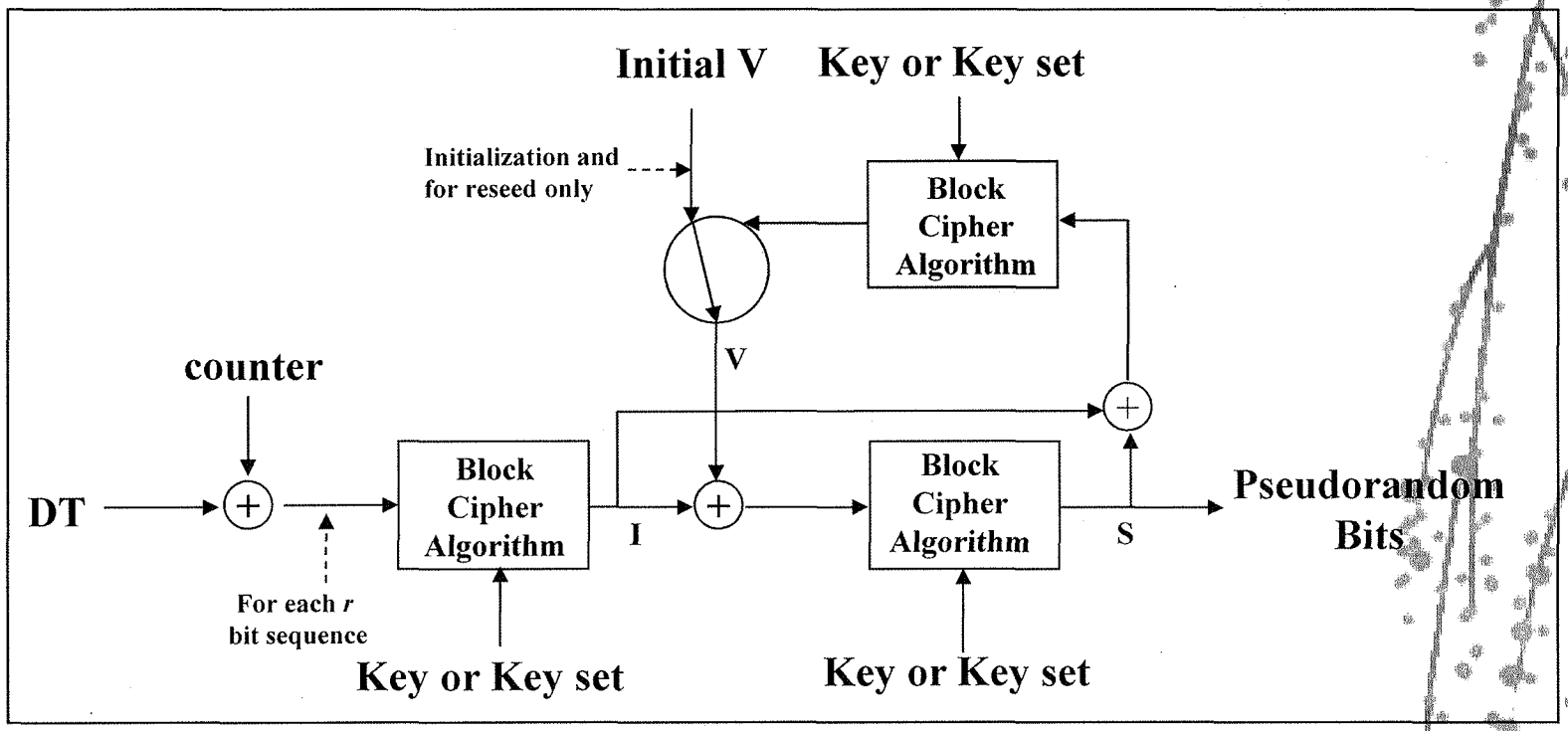


X917DRBG (Contd.)

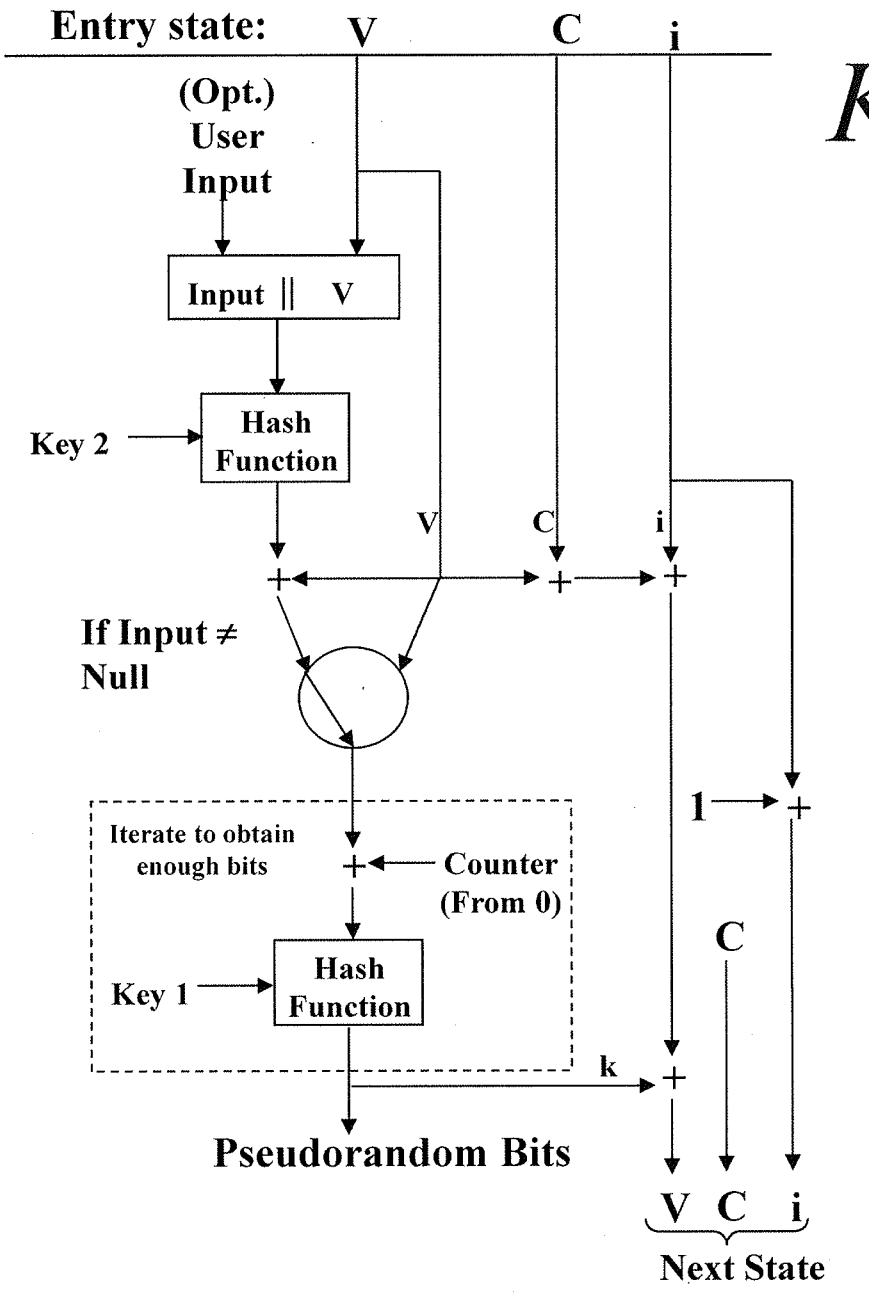
- State: *V*, *strength*, *Key* A, *Key* B, *Key* C
- Strength: Depends on block cipher
  - 2TDEA: 80
  - 3TDEA: 112
  - AES-128: 128
  - AES-192: 192
  - AES=256: 256



X917DRBG



- Uses an Approved symmetric block cipher algorithm (e.g. TDEA or AES).

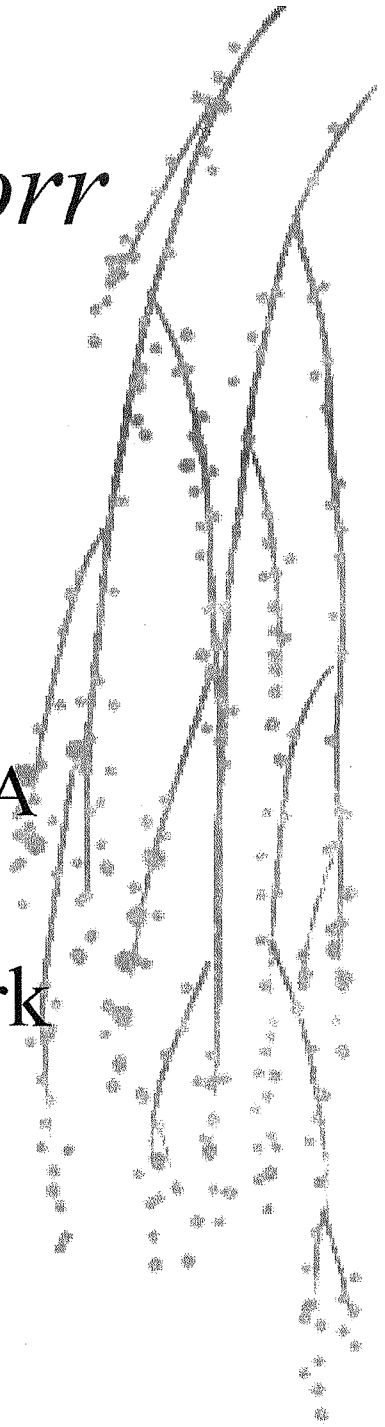

Entry state:
V
C
i
(Opt.)
User
Input
Input || V
Key 2
Hash
Function
V
C
i
If Input ≠
Null
1
Iterate to obtain
enough bits
Counter
(From 0)
C
Key 1
Hash
Function
k
Pseudorandom Bits
V C i
Next State

KeyedHashDRBG

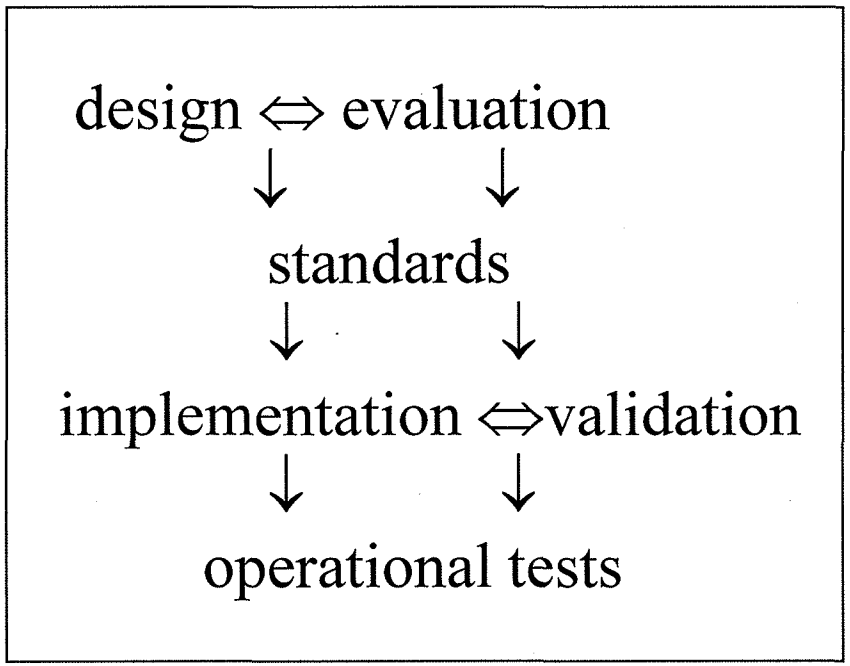
- Uses any Approved hash function
- SEED-SET: *V*, where *vlen* ≥ block size (*N*), and 1-3 *Keys*, where *keylen* ≥ *N*. Each **shall** have entropy ≥ *N*/2.
- Choose hash function in accordance with the desired security strength
- State: *V*, *C*, *i*, *t*, *strength* [, *Keys*]
- Strength: *N*/2

Dual EC and Micali-Schnorr DRBGs

- Based on number theoretic problems
  - Dual EC DRBG: based on elliptic curve logarithm problem
  - Micali-Schnorr DRBG: based on the RSA integer factorization problem
- Basic concept is provided; further work required to complete the specification



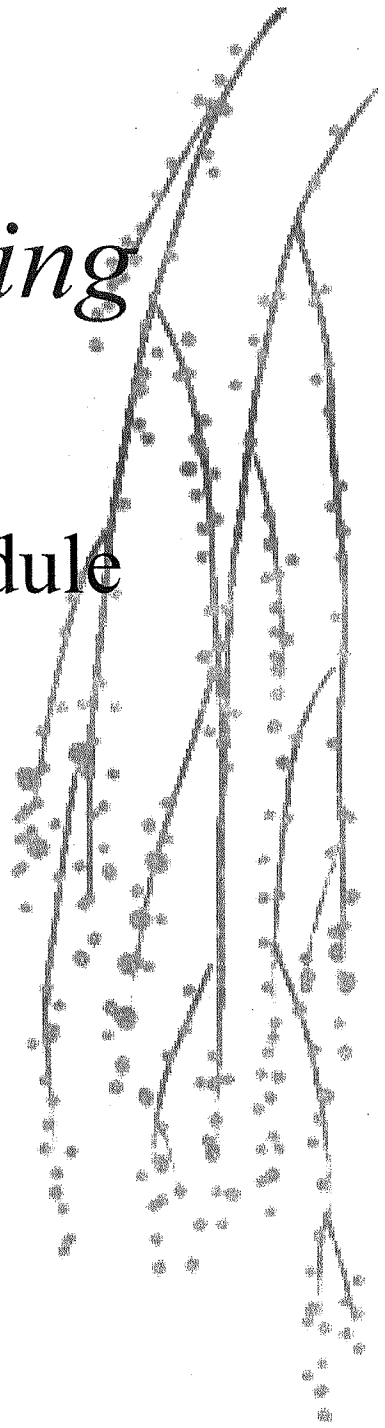
Assurance



- Why is assurance needed?
- Design evaluation
- Implementation validation
- Operational tests

Assurance: Validation Testing

- Implement in a FIPS 140-2 cryptomodule
- DRBG design **shall** include a testing capability



Assurance: Operational Testing

- A DRBG **shall** perform self tests
- Output **shall** be inhibited during testing
- Enter an error state when a test is failed
- Tests:
  - Algorithm Test
  - Software/firmware integrity test
  - Critical functions test
  - Software/firmware load test
  - Manual key entry test