RNG Standard

(Under development by ANSI X9F1)

Elaine Barker

National Institute of Standards and Technology

ebarker@nist.gov

301-975-2911

Who is developing the standard?

- American National Standards Institute (ANSI)
- Financial Services Committee X9
- Security Subcommittee X9F
- Cryptographic Standards and Guidelines working group X9F1
  - Reps. from the financial community, private industry, the U.S. and Canadian govt.
- Editor: NIST

Organization of the Standard

- Being developed in five parts:
  - Overview and Basic Principles
  - Deterministic RBGs Based on Hash Functions
  - Deterministic RBGs Based on Block Ciphers
  - Deterministic RBGs based on Hard Problems
  - Non-Deterministic RBGs

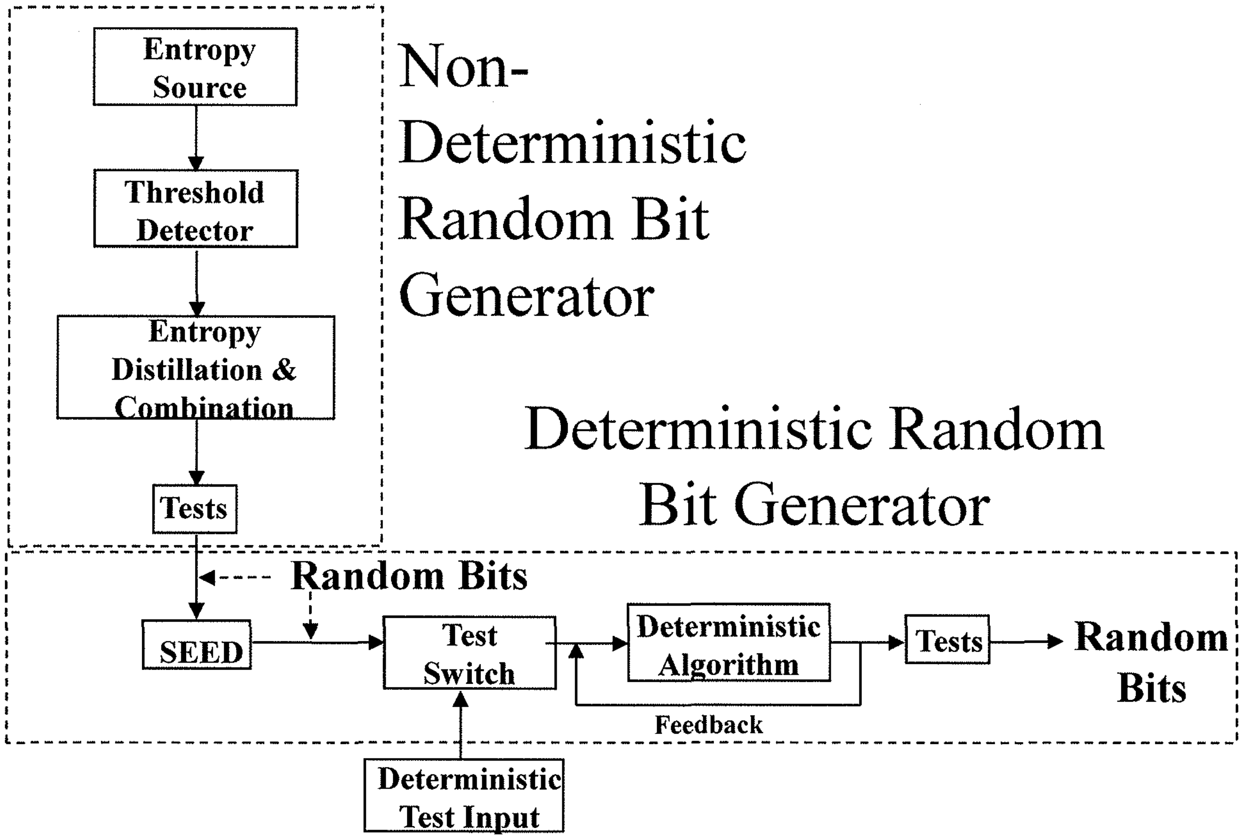
Overview and Basic Principles

- Functional Model
- Top Level Security Requirements (for RBG output, properties and operation)
- RBG Functional Requirements
- Deterministic RBGs
- Non-deterministic RBGs
- Hybrid RBGs

Overview and Basic Principles (contd.)

- Using Multiple RBGs
- Producing Random Numbers from Random Bits
- General Implementation Issues
- Testing
- Appendix: Security Considerations

Functional Model



Entropy Source

Threshold Detector

Entropy Distillation & Combination

Tests

Non-Deterministic Random Bit Generator

Deterministic Random Bit Generator

Random Bits

SEED

Test Switch

Deterministic Algorithm

Tests

Random Bits

Feedback

Deterministic Test Input

Top Level Security Requirements

- Requirements for RBG Output
  - Indistinguishable
  - Pass statistical tests
  - Outputs should appear to be independent
  - Infeasible to exploit repeats
  - Forward and/or backward secrecy provided

Top Level Security Requirements (contd.)

- Requirements for RBG properties and Operation
  - Probability of misbehavior must be small
  - Free from influence or observation
  - Protected consistent with use and output sensitivity
  - No generation of bits without sufficient entropy
  - Be able to recover from loss or compromise of entropy

Top Level Security Requirements (contd.)

- Requirements for RBG properties and Operation (contd.)
  - Forward and/or backward secrecy, as required
  - Verifiable, if required.

Functional Requirements

- Requirements for all RBGs
  - Satisfy the appropriate top-level requirements
  - Design evidence to support all security requirements
  - Verifiable implementation
  - Capable of supporting forward and backward secrecy

Functional Requirements (contd.)

Samples of Functional Requirements

- Entropy source:
  - Based on well established physical principles or behavior
  - Entropy rate must be estimable or self-regulating
  - Free from influence and observation
  - Multiple sources are desirable
  - Degradation of entropy source must be detectable

Functional Requirements (contd.)

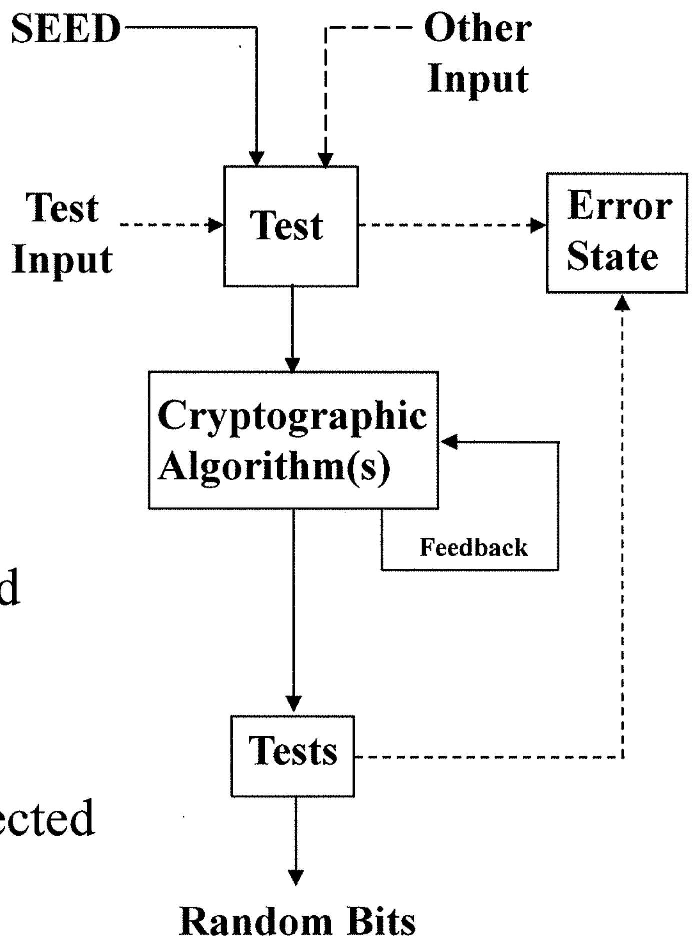
- Samples of Functional Requirements (contd.)
  - Output Function
    - Verifiable
    - Inhibited until sufficient entropy is available
    - Inhibited during testing
    - Depend upon all internal entropy
    - Resistant to producing chosen output.
    - Protect the internal state

Functional Requirements (contd.)

- Samples of Functional Requirements (contd.)
  - Output Function (contd.)
    - Resist observation and analysis
    - Changing one input bit changes half of output bits

Deterministic RBGs

- Model
  - Advantages
    - Often quite fast
    - Reproducible
    - Portable
  - Disadvantages
    - Only as unpredictable as the seed
    - Algs. theoretically repeat
    - Predictable if seed is known
    - Seed & other info. must be protected



SEED

Other Input

Test Input

Test

Error State

Cryptographic Algorithm(s)

Feedback

Tests

Random Bits

Seeds and Reseeding

- Obtained from an Approved Non-deterministic RBG
- An *n* bit seed should have *n* bits of entropy
- Different seeds for different types of data
- Seed size & reseeding determined by the deterministic algorithm
- Seeds handled same as target data
- Consecutive seeds not equal

Other Preset Information

- Key? Counter? Date/time value?
  - Key
    - Each bit independent of seeds & other key bits
    - Generated by an Approved NRBG
    - Replaced periodically
    - Different key for different purposes
  - Counter & date/time - never repeat

Other Elements of a Deterministic RBG

- Testing
  - Implementation tests
  - Operational tests
- Error State
  - Enter error state
  - Inhibit output
- Implementation Requirements
  - Seed & internal state are secret, when required

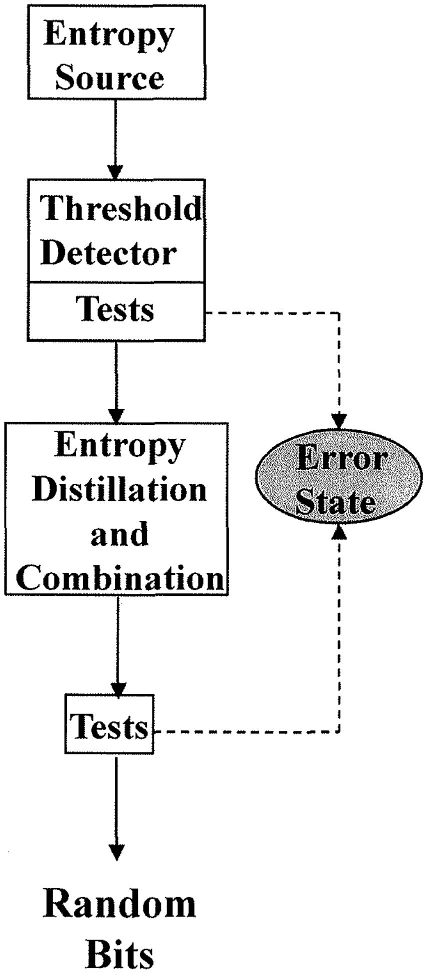
Deterministic RBGs Included (So Far)

- Based on Hash Functions
  - Keyed Hash - (e.g., HMAC)
  - Keyless Hash (e.g., SHA-1)
- Based on Block Ciphers
  - ANSI X9.17 RBG (based on DES)
  - Coming: AES?
- Based on Hard Problems
  - Dual Elliptic Curve
  - Micali-Schnorr

Non-deterministic RBGs

Model

- Advantages
  - Not subject to manipulation, disclosure or predictability if well chosen
- Disadvantages
  - Output may be too slow
  - Could produce deviation from randomness
  - Could fail to repeating values
  - Difficult to validate the design



Non-deterministic RBG Elements

- Entropy Source
  - Numerous possible sources
  - Use combinations of sources?
  - Inherent entropy not subject to influence or observation
- Threshold Detector
  - Reacts to the output of the entropy source
  - Each entropy source uses a separate “pool”

Non-deterministic RBG Elements (contd.)

- Entropy Distillation and Combination
  - Combines & de-skews the output from the threshold detector(s)
- Output Tests
  - Operational tests
- Error State

Other Proposed Topics

- Using Multiple Entropy Sources - TBD
- Design Families? -TBD
- Implementation Criteria - TBD

Testing of Non-deterministic RBGs

- Validation:
  - Use NIST tests for randomness- http://csrc.nist.gov/rng
  - Against criteria in Parts 1 & 5

Testing (contd.)

- Operational Testing (self tests)
  - Based on FIPS 140-2 tests (at power up, on demand & under certain conditions)
  - Output inhibited during testing
  - Test bits ≠ output bits
  - Intervention required when errors encountered

Testing (contd.)

Operational Tests (contd.)

- Tests
  - Deterministic algorithm test, when appropriate
  - Software/firmware integrity test, when appropriate
  - Critical functions test
  - Statistical RBG tests
  - Software/firmware load test, when appropriate
  - Manual key entry test, when appropriate
  - Continuous RBG test

Other RNG Topics

- Hybrid RNGs - TBD
- Using Multiple RBGs - TBD
- Producing RNs from Random Bits

Discussion?