
.82, Part 3
stic Random
erators (DRBGs)
Elaine Barker
NIST
July, 2003

...ey Section Review
...ions 1-5)

- ...eferences
  - ...e list the registry documents?
- Terms and Definitions
- Symbols and Abbreviated Terms
  - Only math notation is included; variables are identified in each DRBG section

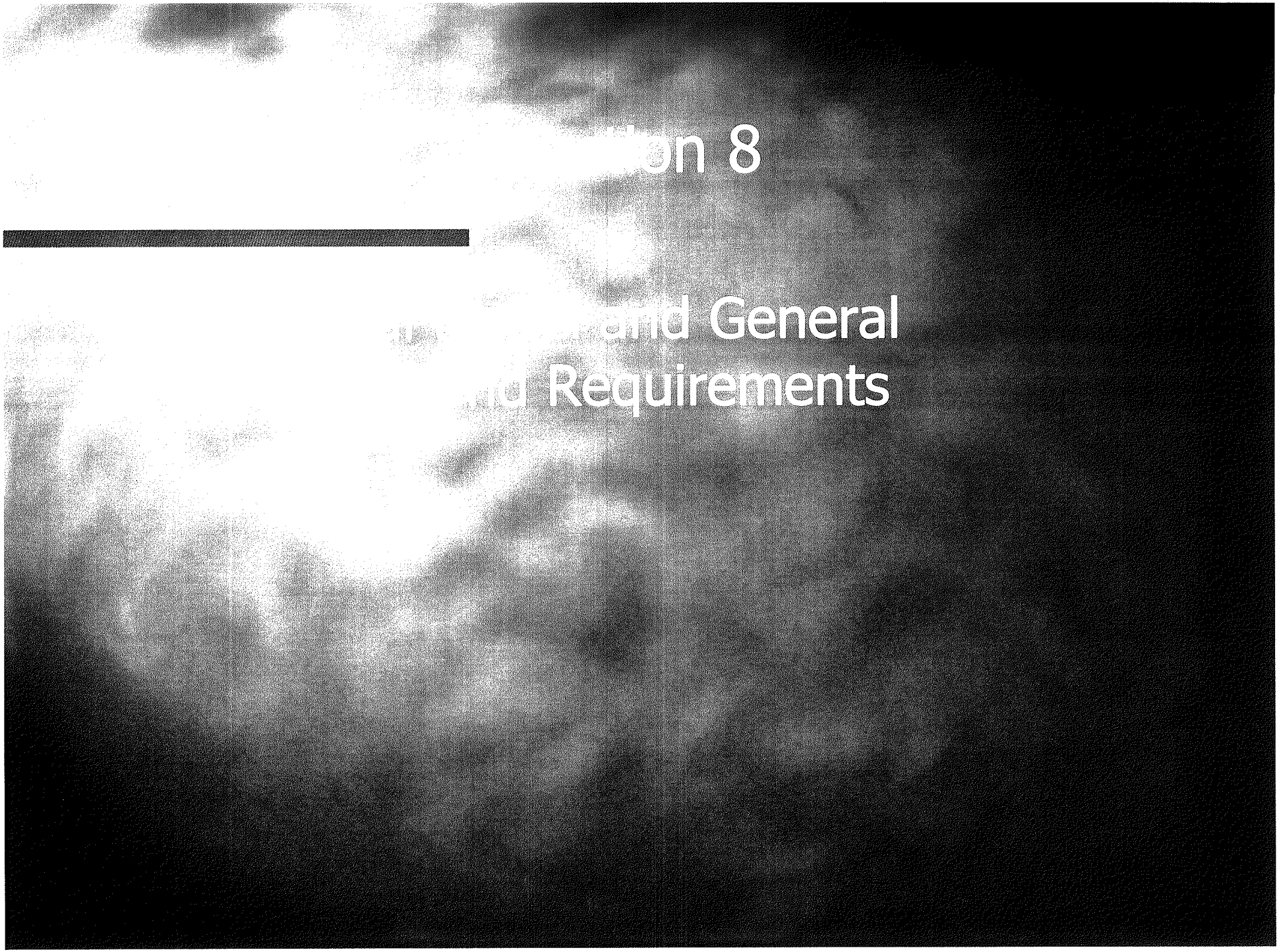
...ions 6-7

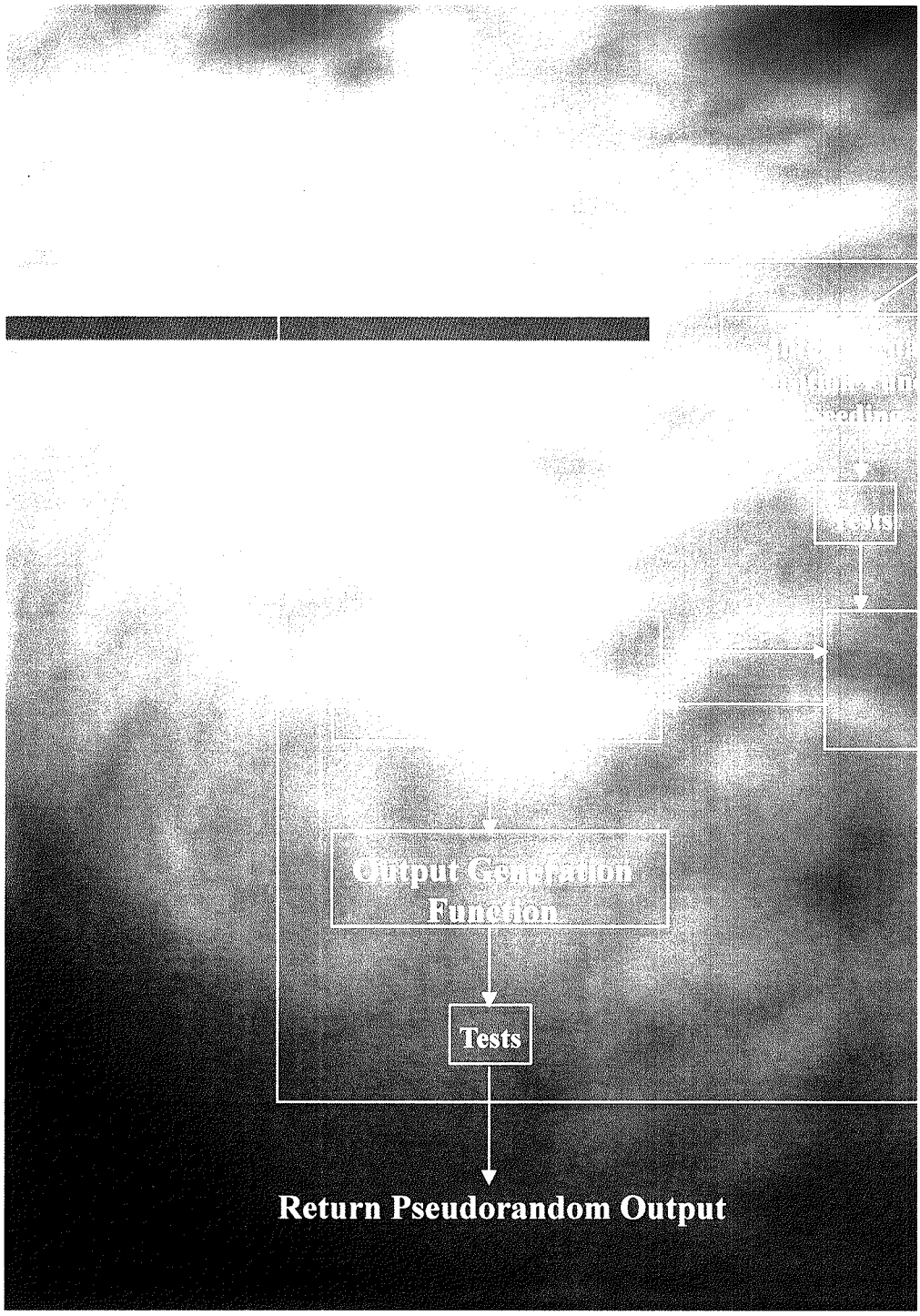
...and Organization

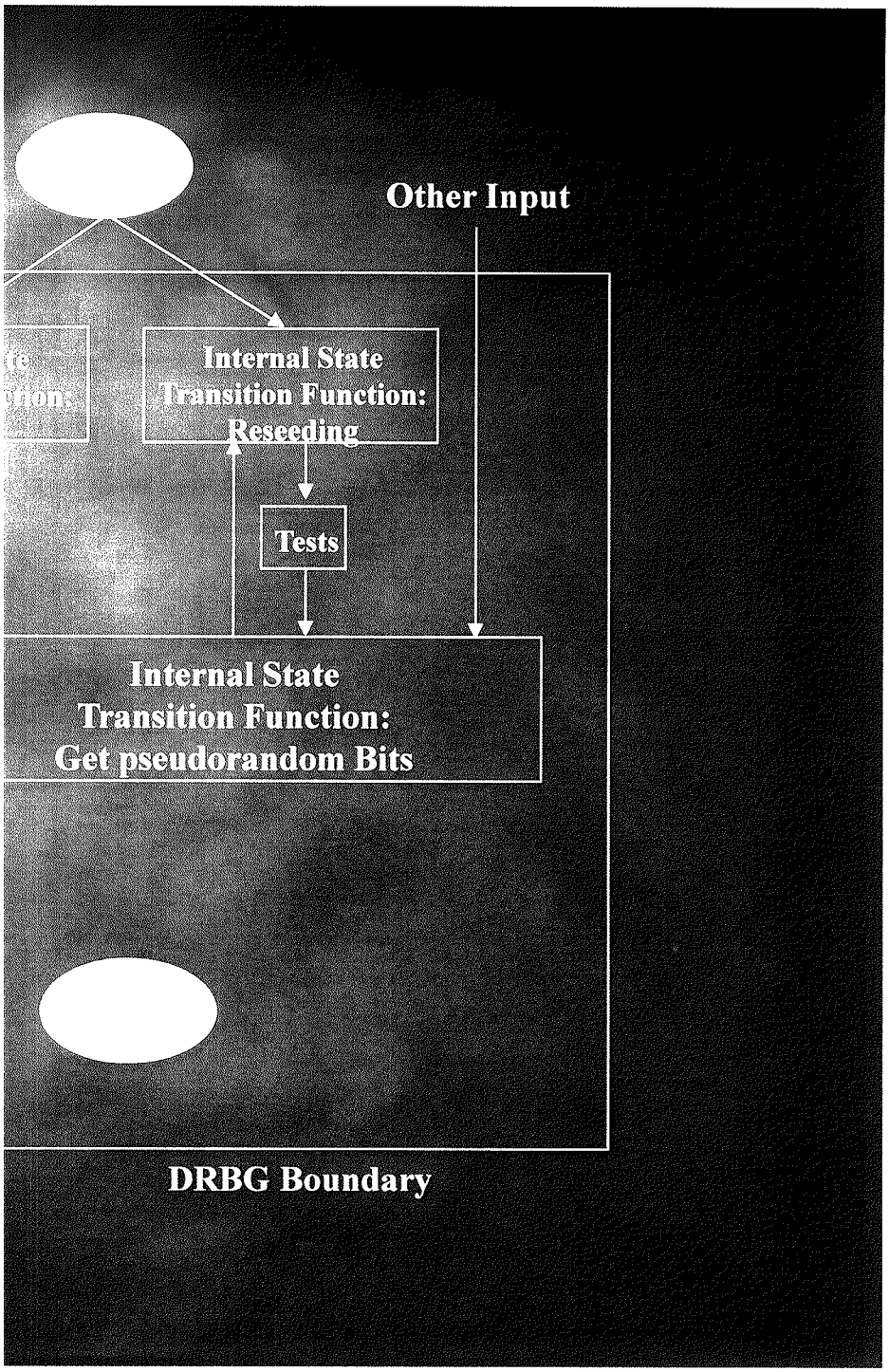
...Objectives and

...les between Part1 1 and 3

- Es... the same requirements and objectives as Part 1.
- RBG Properties and Operation
  - Part 1 testable req. 4 (re protection boundary) not in Part 3.
  - Part 1 testable req. 3 (re backward secrecy) and optional feature 2 (re forward secrecy) are combined as Part 3 objective 3.


on 8
and General
Requirements


Output Generation Function
Tests
Return Pseudorandom Output


Other Input
Internal State
Transition Function:
Reseeding
Tests
Internal State
Transition Function:
Get pseudorandom Bits
DRBG Boundary

...n 8 (Contd.)

...nce of testable req. 1 in Section
...tions?
...ents...
...delines on Part 2, NRBGs)
...er Input?

- Other Input Information
  - Input parameters
  - Time-variant information
  - User input?
- Internal State
  - Depends on the DRBG

...n 8 (Contd.)

...ts... (contd.)

...nsition Functions

...n, Reseeding, & Generate output

...ation Function

- ...test output and secret output separation: Stated as separating test and operational output; do we need to separate output that will be public from output that remains secret (I.e., separate DRBG instances)?

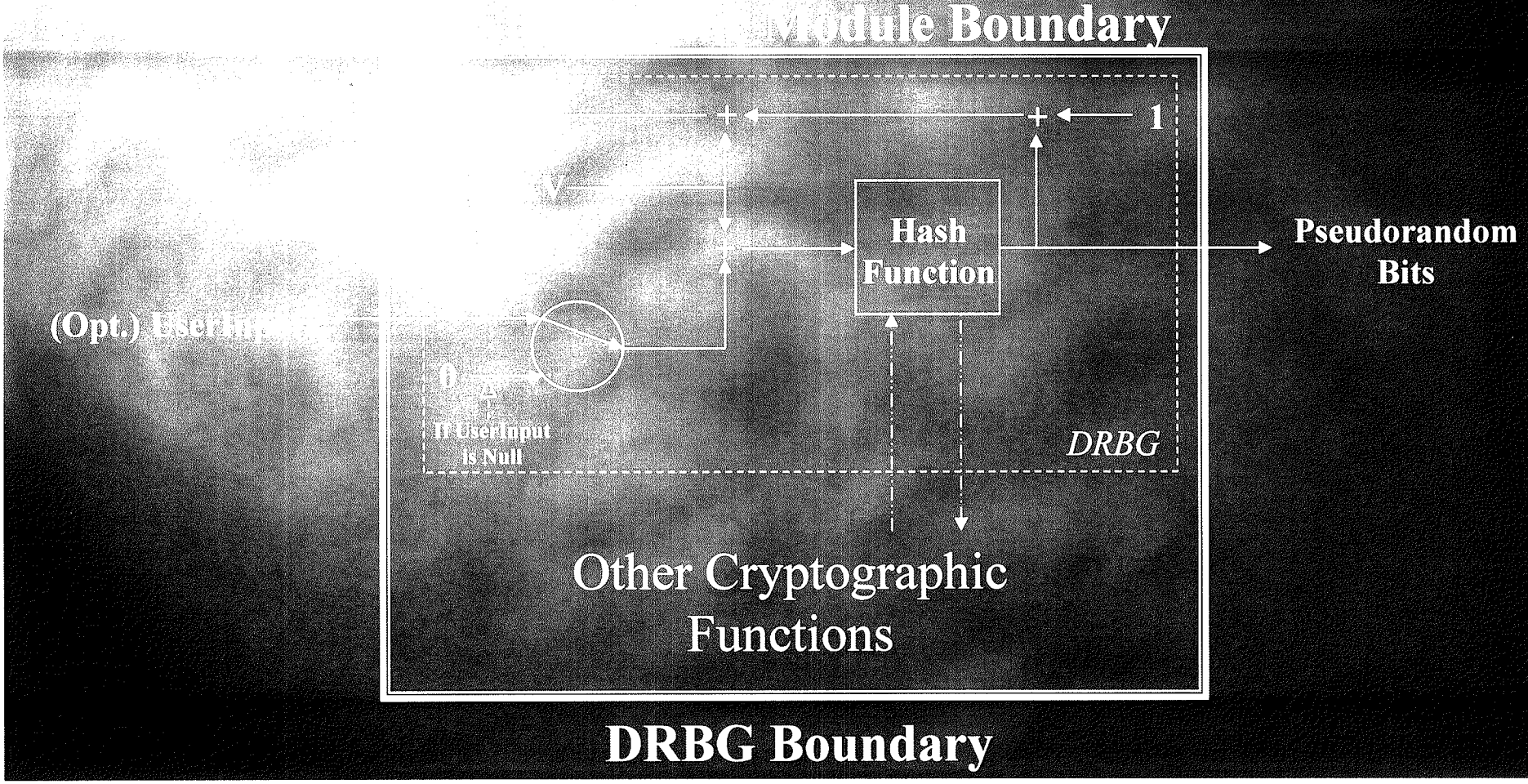
- Support Functions
  - No requirements in Part 1

n 9 – Additional rements

- be in a DRBG boundary
- er inputs **shall not** be
  the boundary
- DRBG boundary **shall** be the same as the cryptomodule boundary or contained within the boundary

irements (Contd.)

ame as cryptomodule):



quirements (Contd.)

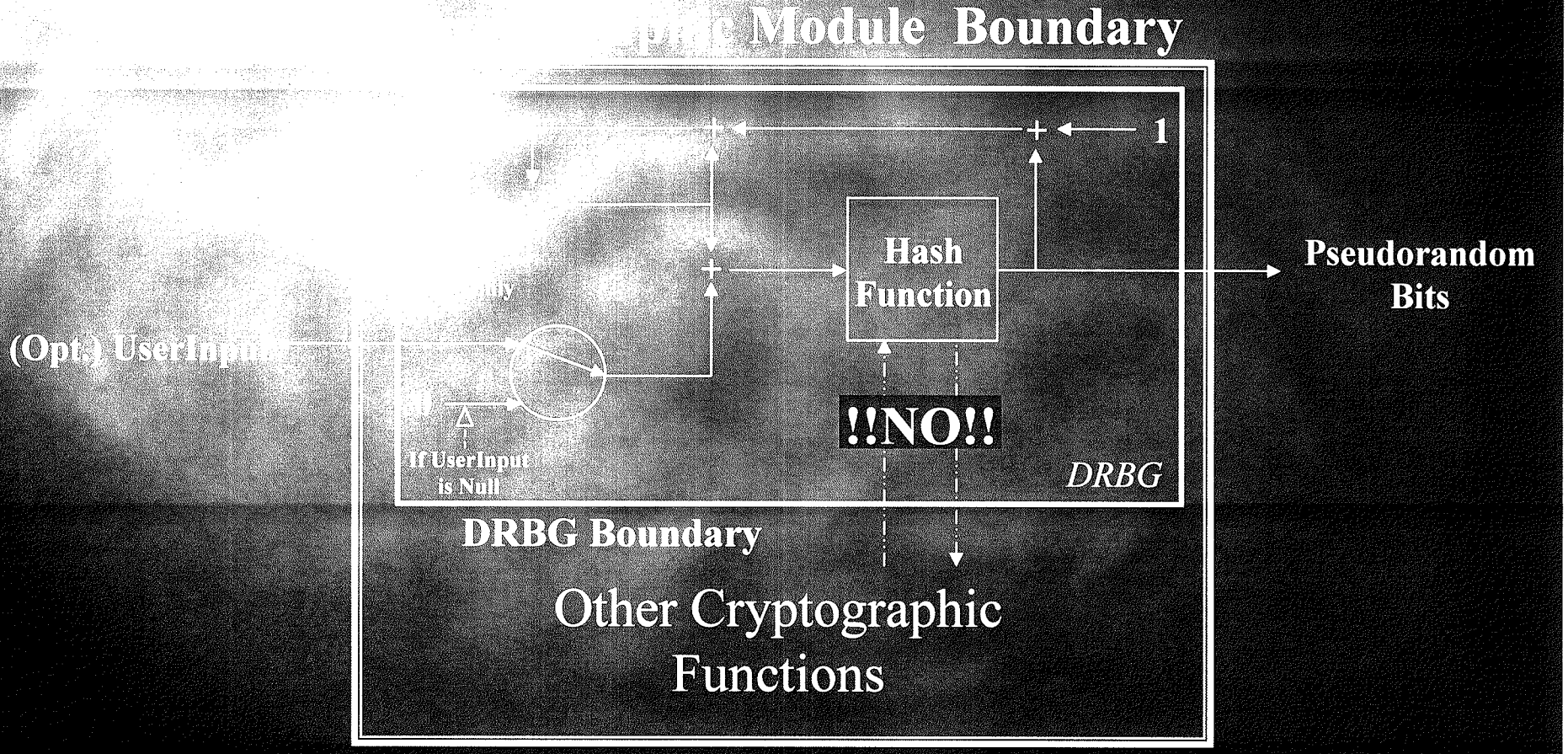
me as cryptomodule)

unctions of the DRBG **may** be used
unctions within the boundary

- Internal state **shall not** be affected by the other crypto functions

equirements (Contd.)

DRBG Boundary within a cryptomodule



quirements (Contd.)

within a cryptomodule
td.):

ents **shall not** be accessible
outside the DRBG boundary
(or within the cryptomodule boundary)

[illegible] Requirements (Contd.)

- [illegible] prior to producing DRBG
  - [illegible] (usage class)
  - [illegible] from the seed(s)
  - [illegible] depending on the DRBG)
  - DRBG [illegible] information
  - Security strength provided
  - Transformation of seed or initial state
- **Shall not** be accessible outside the DRBG
- Transitions on demand; may transition in response to events or continuously

Requirements (Contd.)

- [illegible] prior to the generation of DRBG
  - [illegible] secret
  - **Shall not** be used to generate both secret and public values
  - **Shall not** be reused (including use for another instantiation)
  - Need to add: DRBG **shall not** provide output until a seed(s) is available and initial state is determined
- Seed entropy
  - Needs to be reworded: entropy ≥ security strength

Requirements (Contd.)

reworded: seed size $\geq$ 2 x security

- Entropy source
  - An Approved NRBG, or an Approved DRBG seeded by an Approved NRBG
- Seed Privacy
  - Handling consistent with security required for target data

Requirements (Contd.)

- specified seedlife
- two successive seeds aren't the
- one-way transformation on the
- al state
- Seed separation
  - Different seeds **should** be used for different types of data (i.e., different instances)

quirements (Contd.)

- erated from seeds or provided
  ternal source
- DRBG shall not provide output until a key(s) is available
- Key entropy
  - Need to rewrite: Entropy **shall** be ≥ security strength

Requirements (Contd.)

- support the desired security
- ined from a seed
  - Keys shall be independent of the rest of the initial state determined by the seed (Has this been accomplished in 3BlockCipher_DRBG?)
  - Depending on the DRBG
    - Initial state (including all keys) is determined from a single seed, or
    - Multiple seeds are used to determine different parts of the initial state

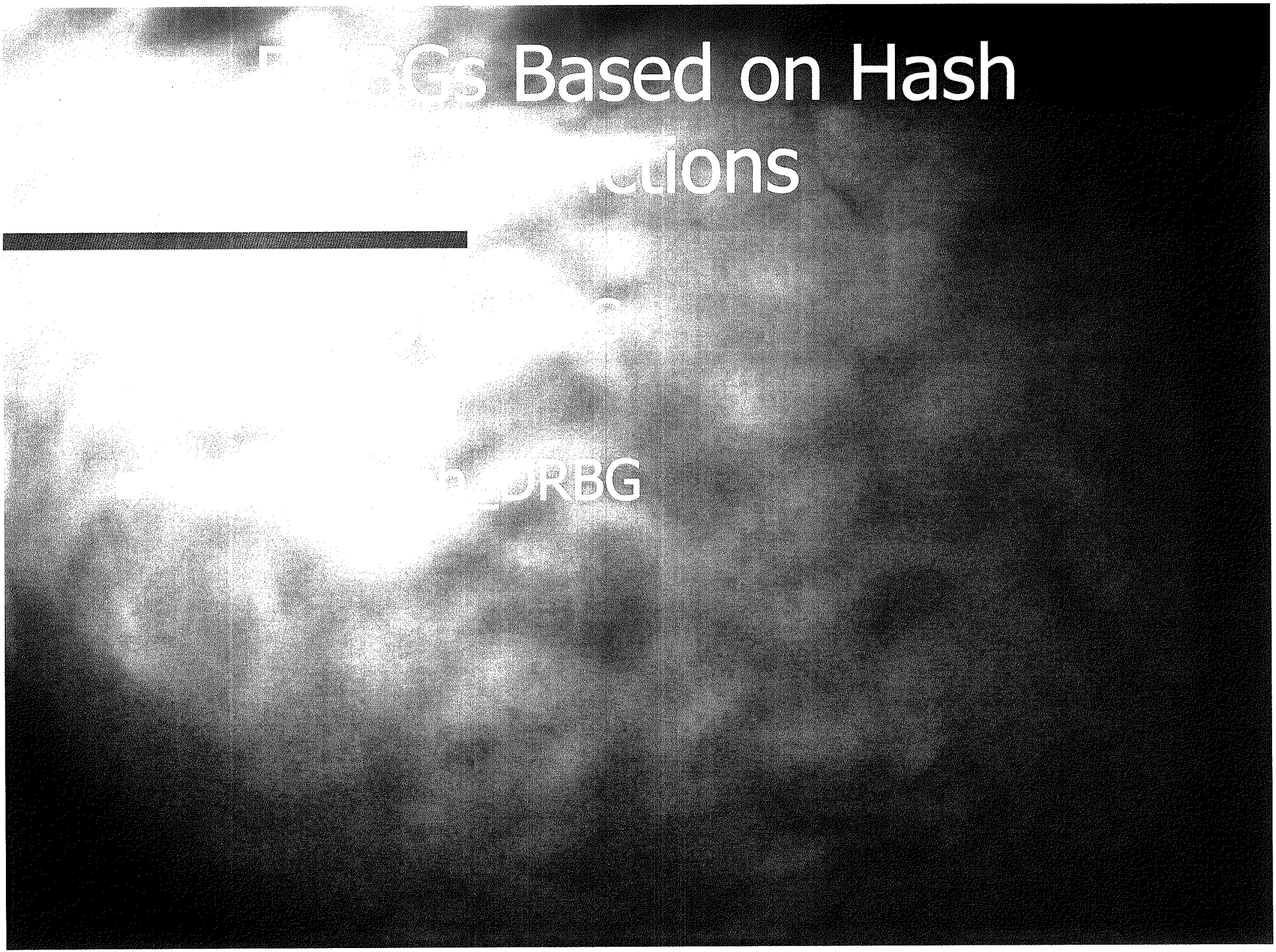
Requirements (Contd.)

- [illegible] an external source
  - [illegible] full entropy (How do we enforce this?)
  - [illegible] protected in accordance with SP 800-57
  - [illegible] the cryptomodule and DRBG
  - [illegible]
  - (can we refer to a NIST special pub?)
- Rekeying
  - Keys **shall** have a finite keylife
- Key separation
  - DRBG keys **shall not** be used for any other purpose
  - Different DRBG instances **should** use different keys (consider problem when using externally provided keys)

Requirements (Contd.)

- alue are arbitrary
- Backward Secrecy
  - observed from outside the DRBG boundary (black box view)
  - From inside the DRBG boundary
    - Each DRBG provides backward secrecy (as viewed from inside the DRBG boundary) – Is this really true?
    - Some level of forward secrecy provided when new entropy provided

DRBGs Based on Hash Functions



Hash_DRBG

- [illegible] for 80 bits of security
- [illegible]te; seed **shall** have entropy ≥
- State: purpose, *V*, initial value *t* for the hash function, seed entropy, seed transformation
- Specifications for initializing and reseeding the state, and generating output
- DRBG strength and attributes: ???
- Reseeding: How often?

Hash_DRBG

- Uses a hash function
- Different security levels
- Uses different hash functions: **Shall** be the same
- Need to instantiate: seed **shall** have entropy ≥ desired security strength
- State: purpose, *V*, *C*, counter, application-specific constant *t*, security strength, transformed seed

RBG (Contd.)

- initializing and
- state, and generating
  - functions be written for the case where multiple hash functions are available?
- DRBG strength and attributes: ???
- Reseeding: How often?

Hash_DRBG

- DRBG with 1-3 keys
  - for V and each different
  - erated key
  - ...ite) seeds **shall** have entropy ≥ desired security strength
- All keys provided externally, or all keys generated internally – reasonable?
  - Can we check the entropy of externally provided keys?
  - Should we allow some keys to be the same (see "*which_keys*" table on page 67)?

DRBG (Contd.)

- C, counter, application-
  , security strength, keys,
  eed and keys
- s for initializing and reseeding the state and generating output
  - Should the functions be written for the case where multiple hash functions are available?
- DRBG strength and attributes: ???
- Reseeding and rekeying: How often?

...ed on Block Ciphers

...pher_DRBG

...of the RNG in X9.17 and

- Uses an approved block cipher (i.e., TDES or AES)
- Uses 3 instances of the same block cipher (including key sizes) – reasonable?

_DRBG (Contd.)

- sets), one for each
  - DES: not allowed
  - bits of key per key set
  - 168 bits of key per key set
  - AES-X: X bits for each key
- Keys (key sets) may all be the same or some (or all) may be different
- All keys **shall** be generated internally (see comment [ebb25]

[redacted]_DRBG (Contd.)

- [redacted] seeds **shall** have [redacted]ired security strength
- [redacted]eys derived from a single seed
  - AES key wrap algorithm adapted as key derivation function (for now)
- State: purpose, *V*, counter, keys or key sets, security strength, transformed seed

DRBG (Contd.)

- initializing and
- state, and generating
- DT has been changed
  - TDES: counter
  - AES: date/time || counter (allow date/time to be a constant?)
- DRBG strength and attributes: ???
- Reseeding and rekeying: How often?

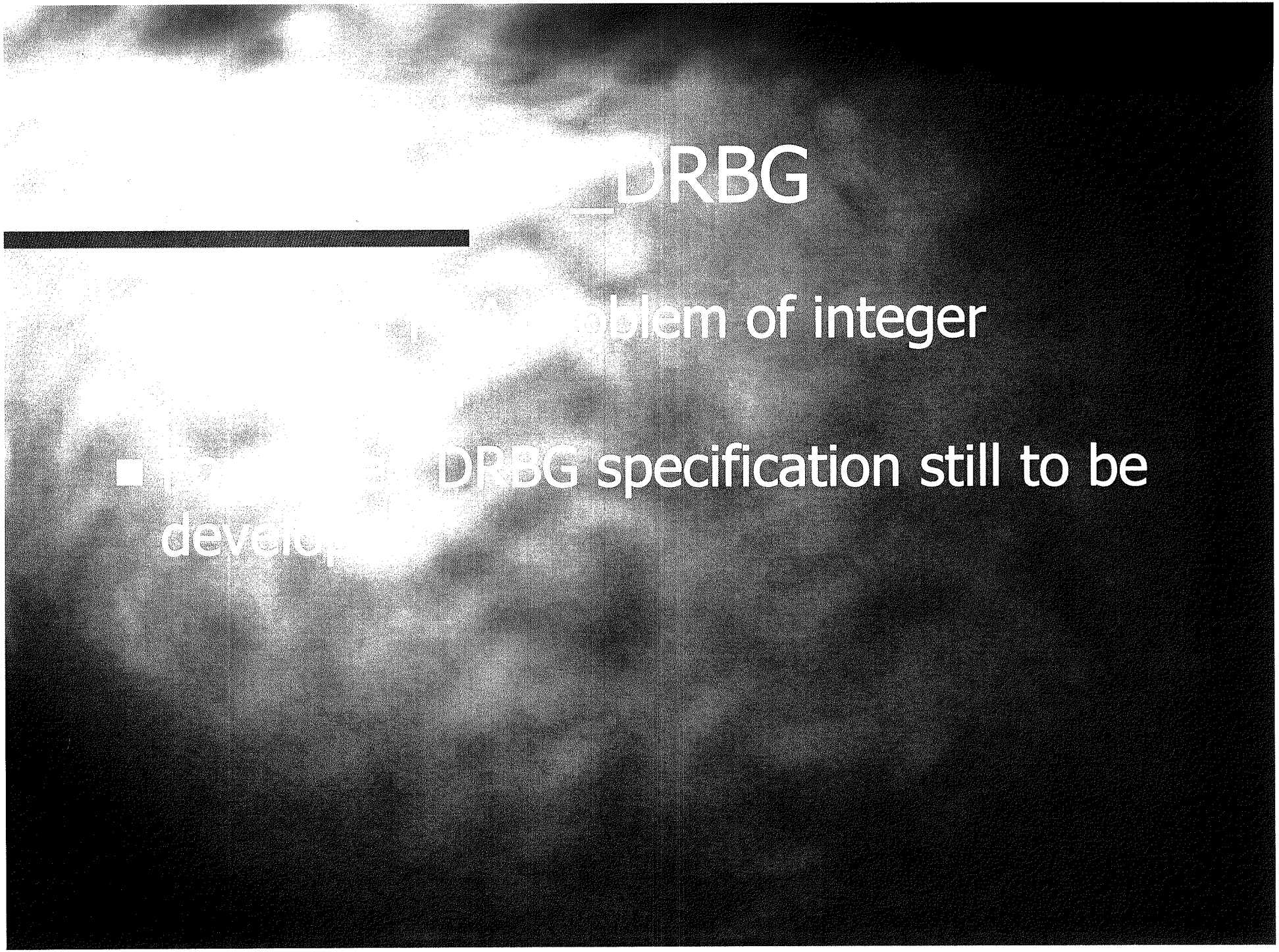
Based on Number ic Problems

_EC_DRBG

- Base ic curve logarithm problem
- Requires 2 points P and Q: P is assigned to G (the generating point) – OK?
- Need to rewrite: seeds **shall** have entropy ≥ desired security strength

DRBG (Contd.)

- prime modulus p,
  meters (includes P), Q,
  ength, transformed seed
- s for initializing and reseeding the state, and generating output
- DRBG strength and attributes: ???
- Reseeding and rekeying: How often?


DRBG
blem of integer
DRBG specification still to be

Assurance

- e evaluated prior to X9.82
  - ing statistical tests)
- ion accuracy may be asserted or,
  - y be validated.
- Validation
  - DRBG **shall** be within a FIPS 140-2 cryptomodule
  - DRBG **shall** be designed to be tested
  - Validation process to be specified in a TG?

ce (Contd.)

ing

erform self tests at power-
and and under various

- DR t **shall** be inhibited during testing
- Results from known-answer tests **shall not** be used as operational DRBG output
- Enter an error state upon test failure
  - DRBG **shall not** output data
  - User intervention required to exit the error state

ces (Contd.)

- gorithm: perform known-answer
- mware integrity test during power up (MACs, digital signatures, EDCs)
- Critical functions tested during power-up and on demand
- Software/firmware load test on all software and firmware that is loaded from an external source

ces (Contd.)

- entry test
  - an EDC (at least 16 bits) or provide
  - ntries (Is an EDC good enough?)
- Continuous RBG test (Is this appropriate for a DRBG?)