Pseudorandom number generator

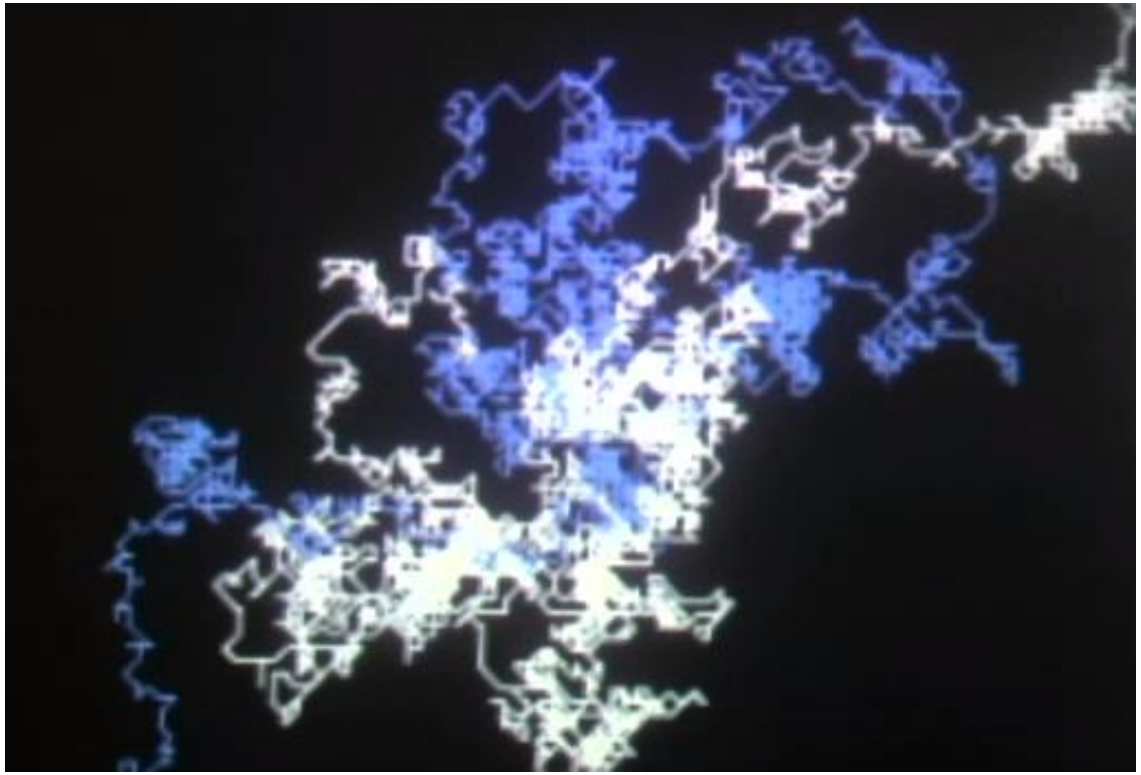
Generátor pseudonáhodných čísel

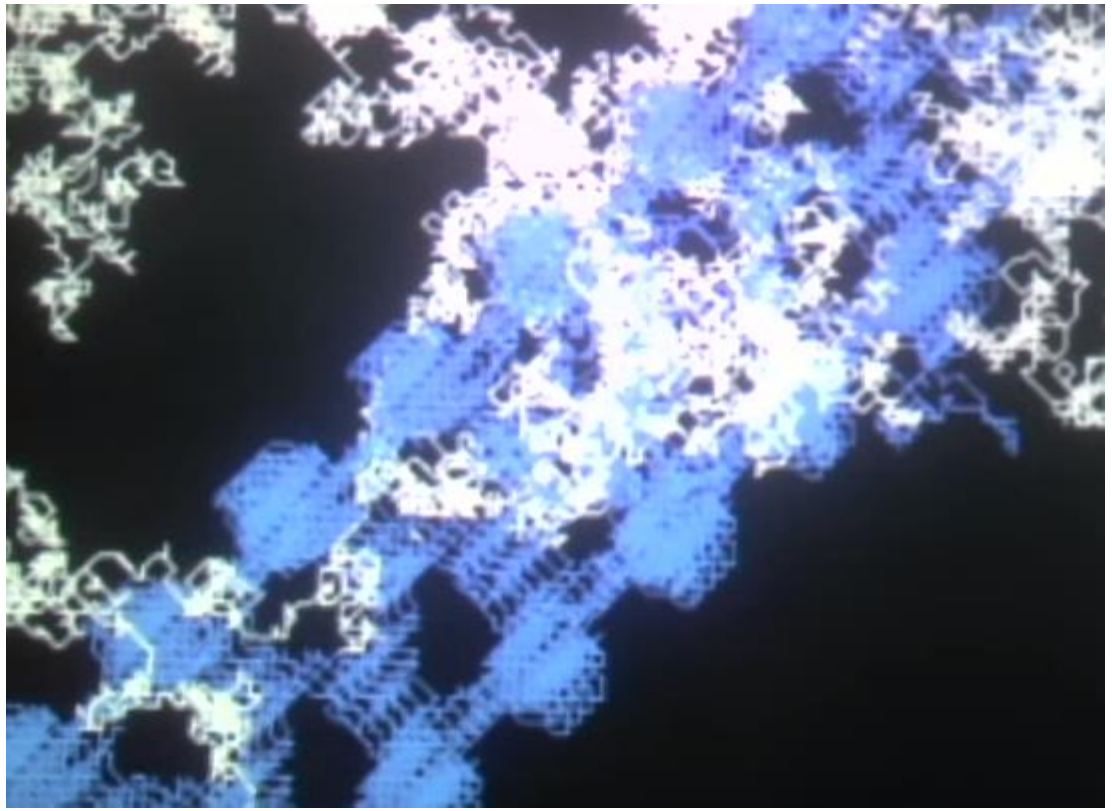
Non-deterministic random bit generators (NRBGs)

- Nezávislosť na fyzikálnych procesoch

Deterministic random bit generators (DRBGs)

- Používanie algoritmu
- Keď vstupný vektor je utajený a algoritmus dobre vytvorený, výsledok bude nepredvídateľný vzhľadom na bezpečnosť algoritmu





1. Pseudorandom number generator

John von Neumann- návrh vodíkovej bomby

- Počítač Eniac
- Procesy pri jadrovej fúzii
- Metóda prostredných štvorcov:

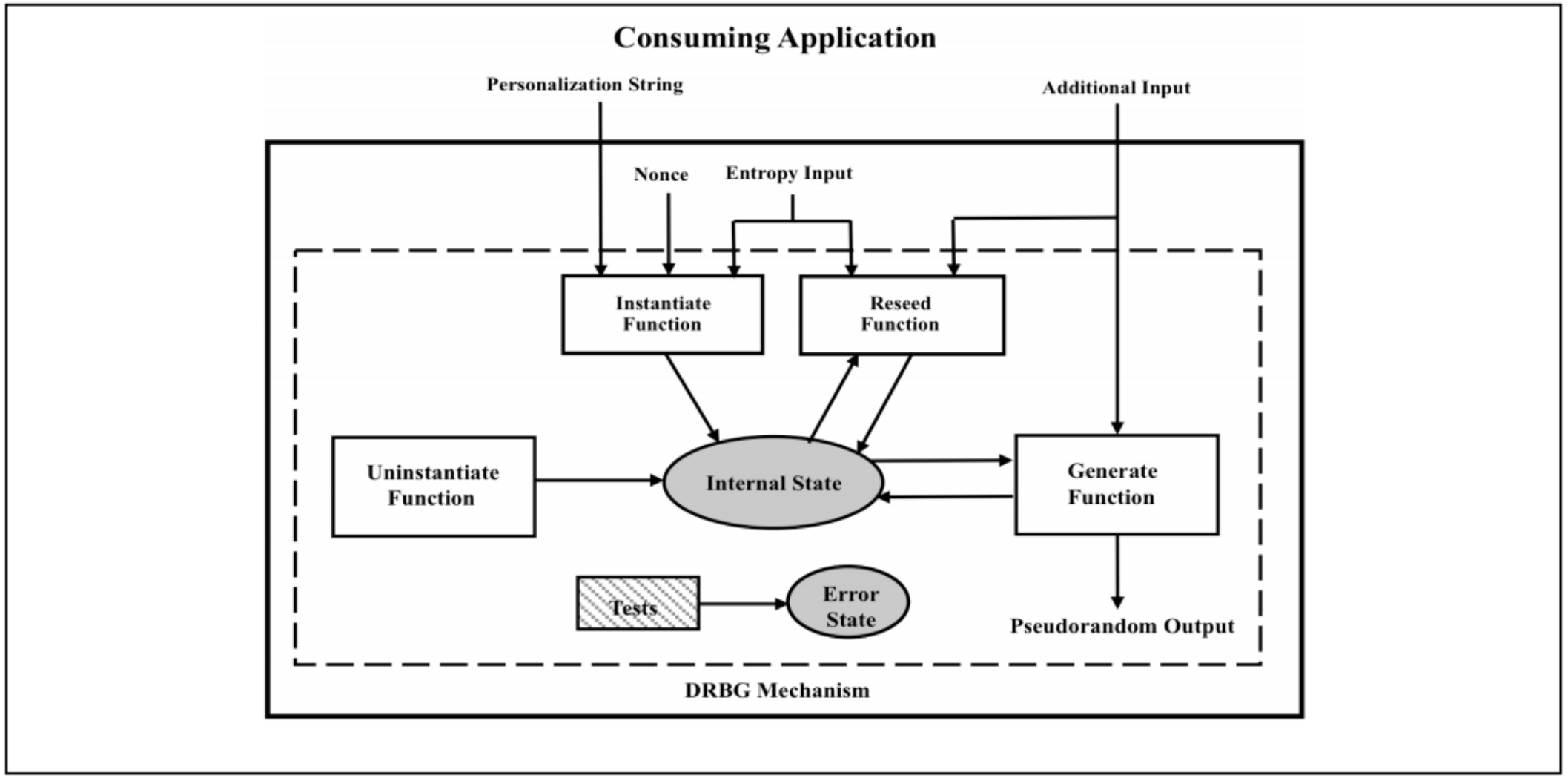
121*121 -> 1**464**1

464*464 -> 21**529**6

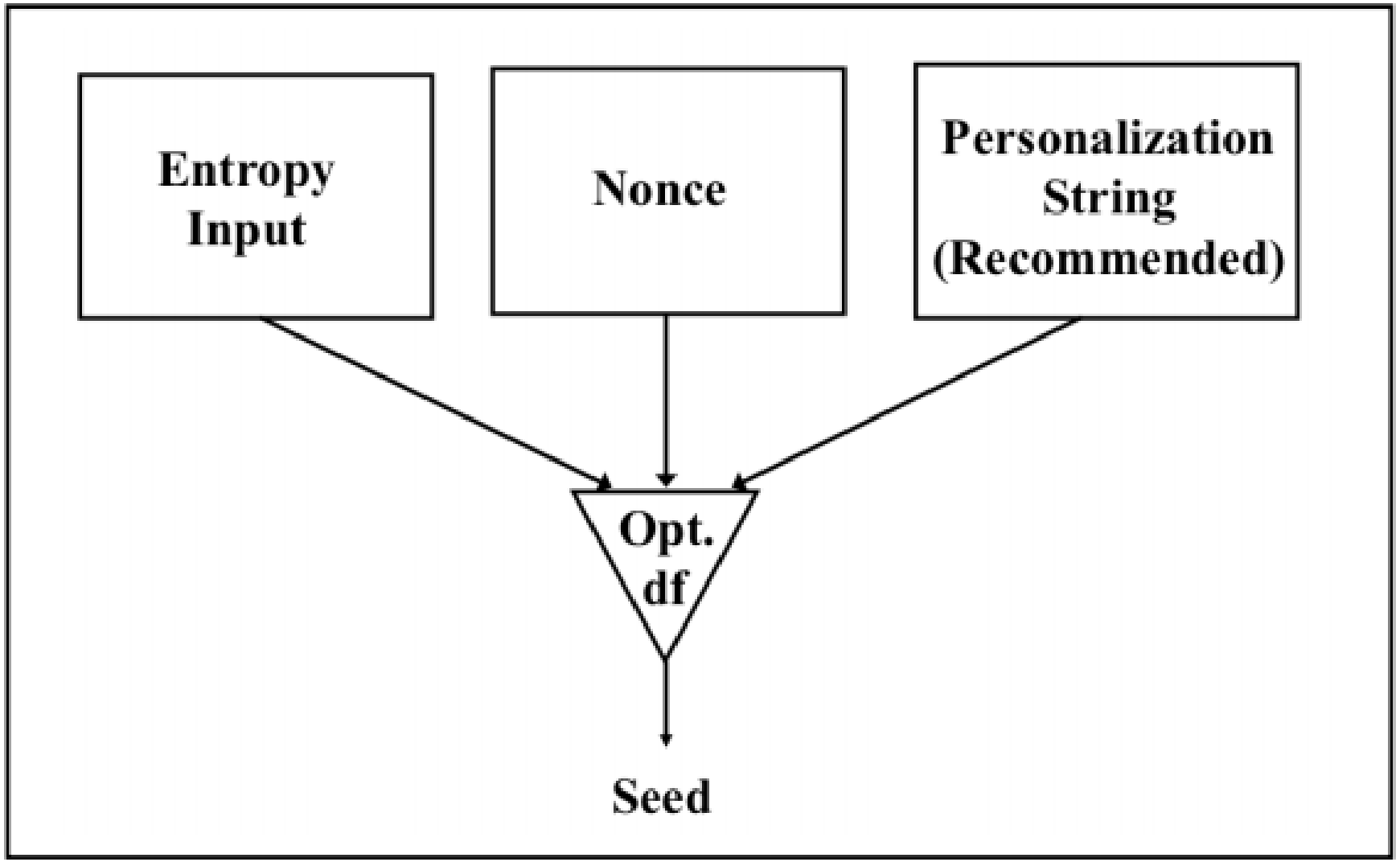
529*529 - > 27**984**1

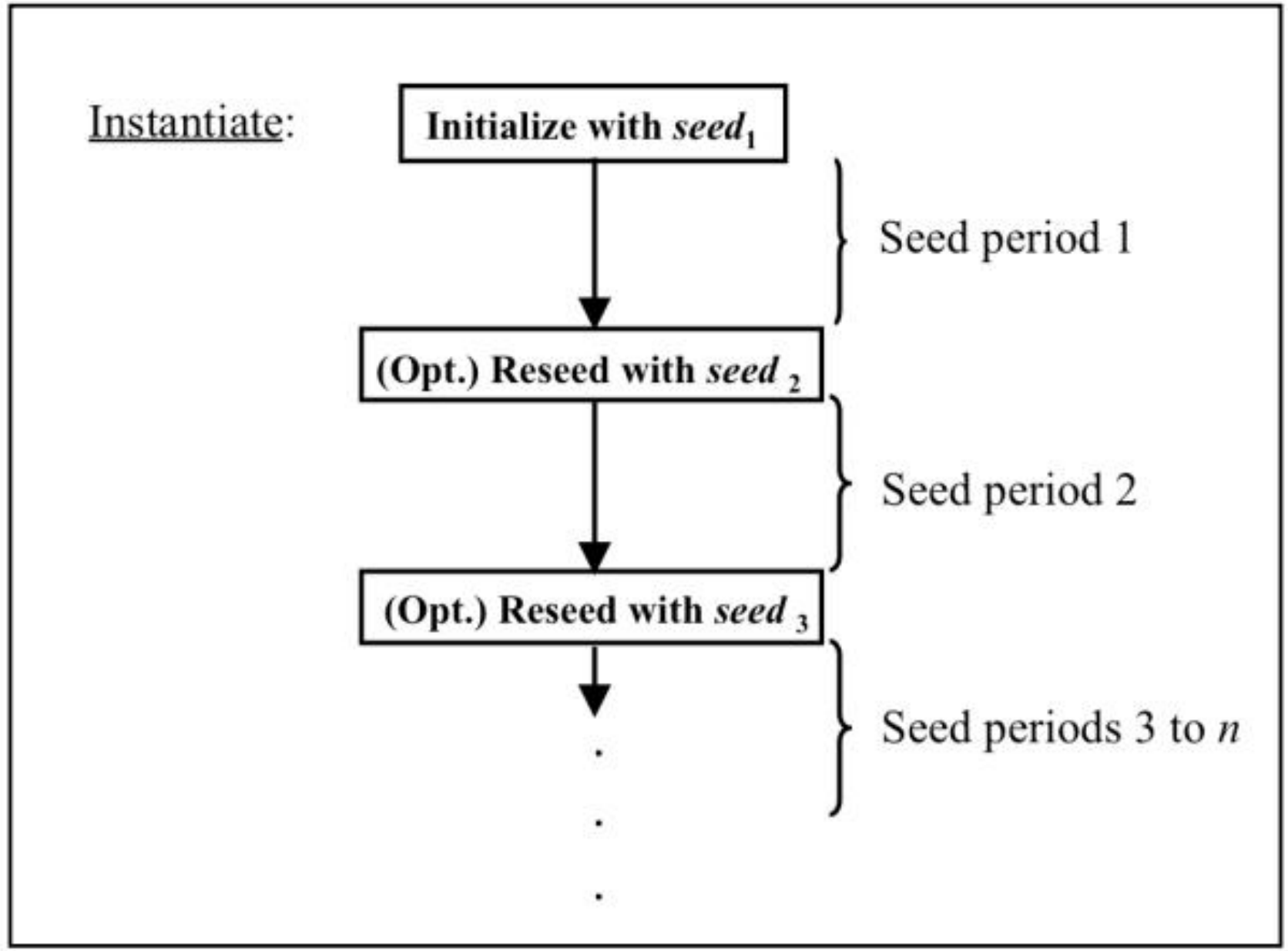
->464529984...

Funkčný model DRBG



Seed


Entropy Input
Nonce
Personalization String (Recommended)
Opt. df
Seed


Instantiate:
Initialize with seed1
(Opt.) Reseed with seed 2
(Opt.) Reseed with seed 3
Seed period 1
Seed period 2
Seed periods 3 to n

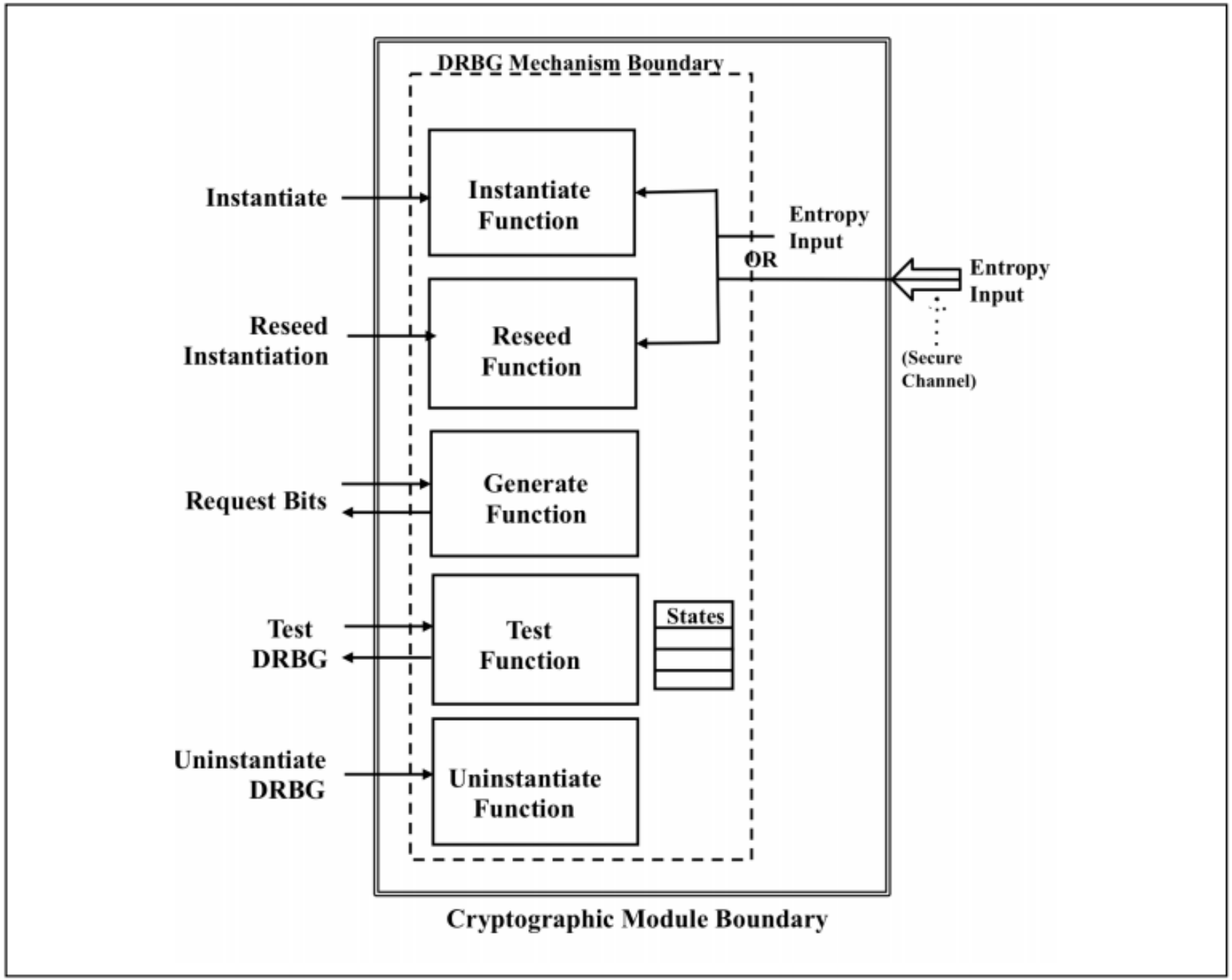
Maximum Designed Security Strength	**112**	**128**	**192**	**256**
Possible Instantiated Security Strengths	112	112, 128	112, 128, 192	112, 128, 192, 256

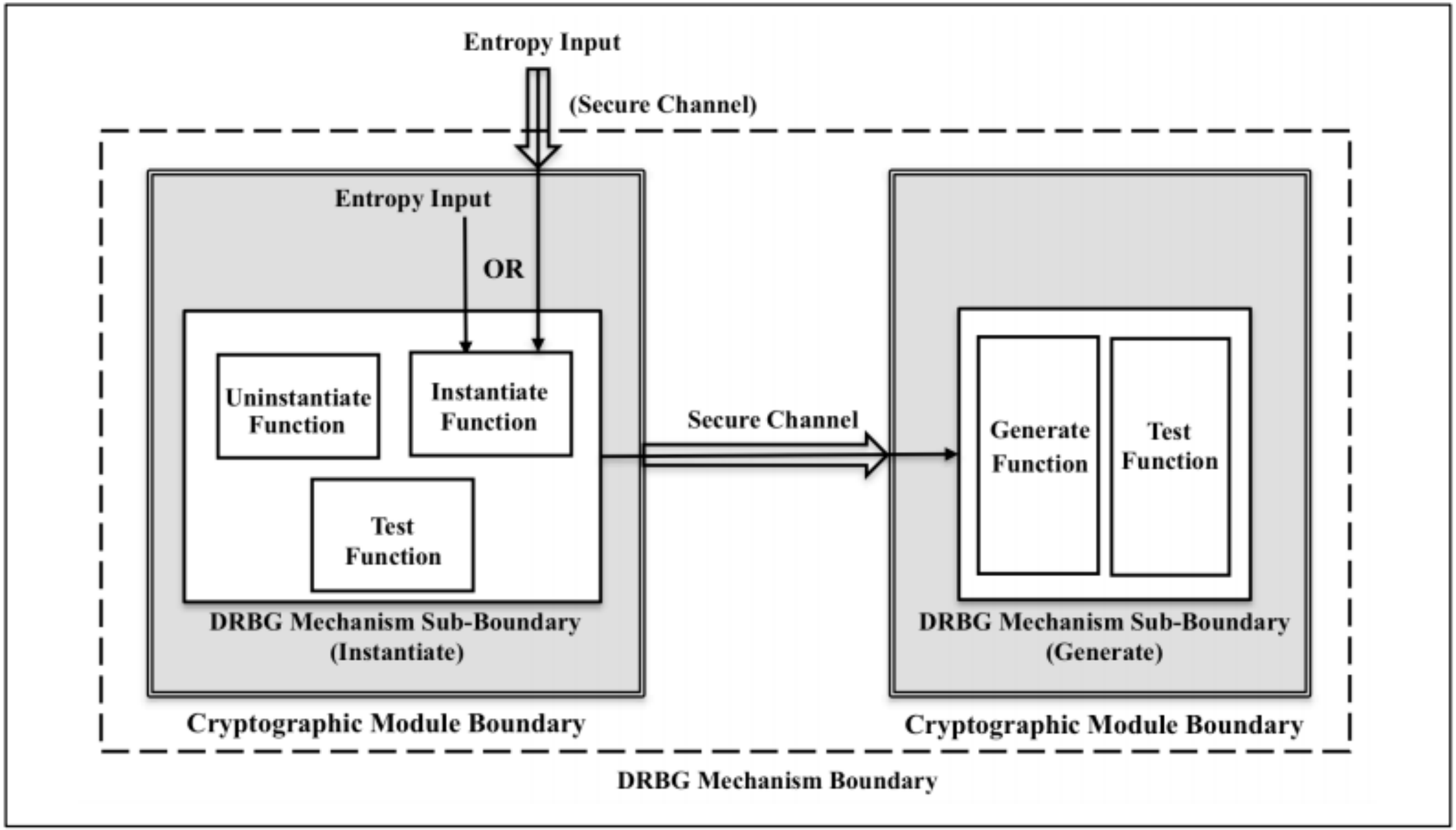
Entropy input

$$\sum entropie = požadovaná\ entropia$$

$$\sum entropie < požadovaná\ entropia$$

$$\sum entropie > požadovaná\ entropia$$


DRBG Mechanism Boundary
Instantiate
Instantiate Function
Entropy Input
OR
Entropy Input
Reseed Instantiation
Reseed Function
(Secure Channel)
Request Bits
Generate Function
Test DRBG
Test Function
States
Uninstantiate DRBG
Uninstantiate Function
Cryptographic Module Boundary


Entropy Input
(Secure Channel)
Entropy Input
OR
Uninstantiate Function
Instantiate Function
Test Function
DRBG Mechanism Sub-Boundary (Instantiate)
Cryptographic Module Boundary
Secure Channel
Generate Function
Test Function
DRBG Mechanism Sub-Boundary (Generate)
Cryptographic Module Boundary
DRBG Mechanism Boundary

Rozhodovanie DRBG

1. Overenie platnosti vstupných parametrov
2. Určenie sily zabezpečenia
3. Získanie entropy vstupu
4. Získanie nonce
5. Určenie počiatočného vnútorného stavu

Get_entropy_input

(status, entropy_input) = **Get_entropy_input** (min_entropy, min_ length, max_ length, prediction_resistance_request)

Chybové hlásenia:

ERROR_FLAG

CATASTROPHIC_ERROR_FLAG.

Instantiate_function
(requested_instantiation_security_strength,
prediction_resistance_flag,
personalization_string)

1. If requested_instantiation_security_strength > highest_supported_security_strength, then return (ERROR_FLAG, Invalid).

2. If prediction_resistance_flag is set, and prediction resistance is not supported, then return (ERROR_FLAG, Invalid).

3. If the length of the personalization_string > max_personalization_string_length, return (ERROR_FLAG, Invalid).

4. Set security_strength to the lowest security strength greater than or equal to requested_instantiation_security_strength from the set {112, 128, 192, 256}.

5. Null step.

Comment: Obtain the entropy input.

6. (status, entropy_input) = Get_entropy_input (security_strength, min_length, max_length, prediction_resistance_request).

7. If (status ≠ SUCCESS), return (status, Invalid).

8. Obtain a nonce.

9. initial_working_state = Instantiate_algorithm (entropy_input, nonce, personalization_string, security_strength).

10. Get a state_handle for a currently empty internal state. If an empty internal state cannot be found, return (ERROR_FLAG, Invalid).

11. Set the internal state for the new instantiation

12. Return (SUCCESS, state_handle).

Generate_function

(state_handle, requested_number_of_bits, requested_security_strength, prediction_resistance_request, additional_input)

1.Using state_handle, obtain the current internal state for the instantiation. If state_handle indicates an invalid or unused internal state, then return (ERROR_FLAG, Null).

2. If requested_number_of_bits > max_number_of_bits_per_request, then return (ERROR_FLAG, Null).

3. If requested_security_strength > the security_strength indicated in the internal state, then return (ERROR_FLAG, Null).

4. If the length of the additional_input > max_additional_input_length, then return (ERROR_FLAG, Null).

5. If prediction_resistance_request is set, and prediction_resistance_flag is not set, then return (ERROR_FLAG, Null)

6. Clear the reseed_required_flag

7. If reseed_required_flag is set, or if prediction_resistance_request is set, then

7.1 status = Reseed_function (state_handle, prediction_resistance_request, additional_input).

7.2 If (status ≠ SUCCESS), then return (status, Null).

7.3 Using state_handle, obtain the new internal state.

7.4 additional_input = the Null string.

7.5 Clear the reseed_required_flag.

8. (status, pseudorandom_bits, new_working_state) = Generate_algorithm (working_state, requested_number_of_bits, additional_input).

9. If status indicates that a reseed is required before the requested bits can be generated, then

9.1 Set the reseed_required_flag.

9.2 If the prediction_resistance_flag is set, then set the prediction_resistance request indication.

9.3 Go to step 7.

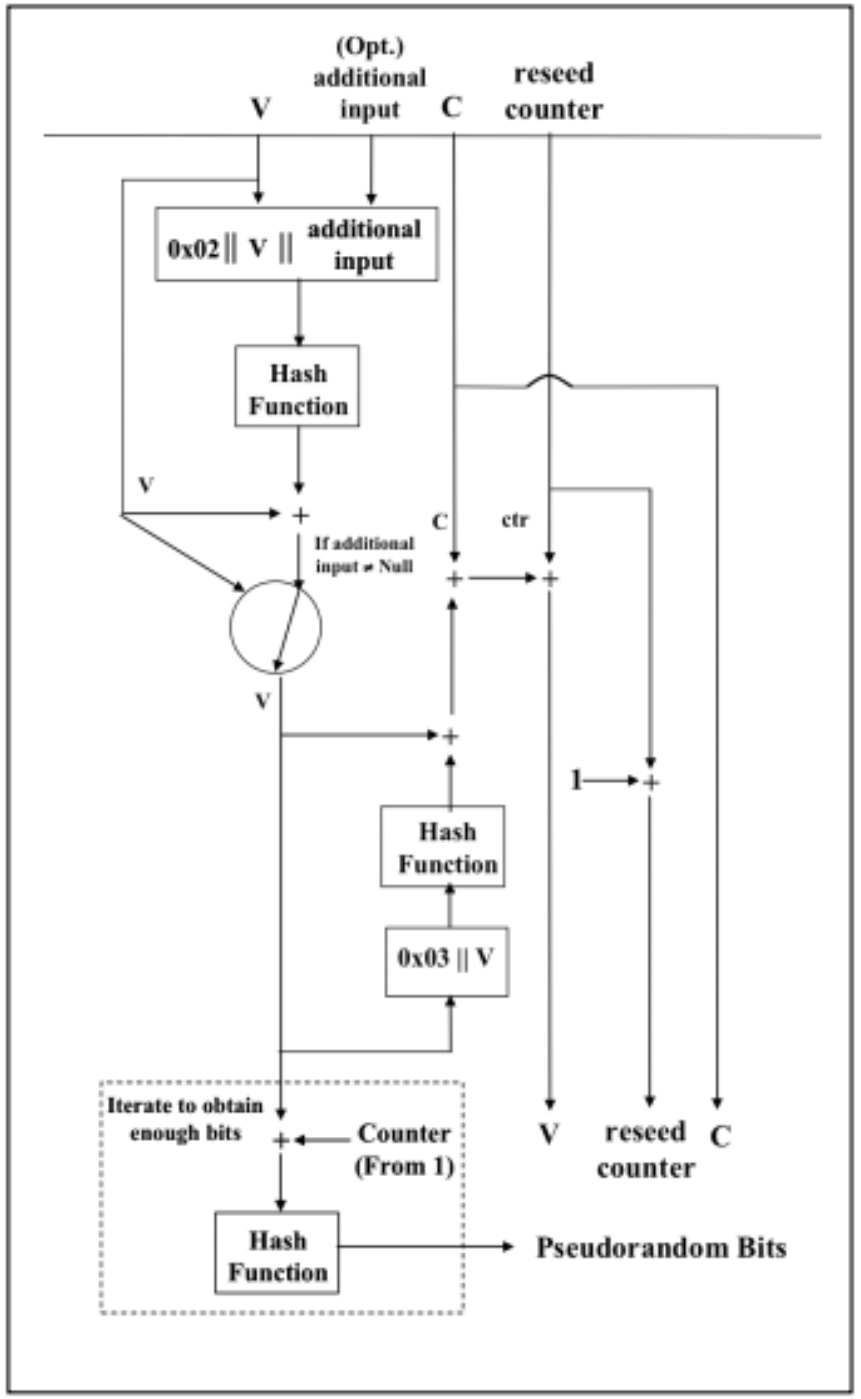
10. Replace the old working_state in the internal state of the DRBG instantiation (e.g., as indicated by state_handle) with the values of new_working_state.

11. Return (SUCCESS, pseudorandom_bits).

HASH rozhodovací proces

1. seed_material = entropy_input || nonce || personalization_string.
2. seed = Hash_df (seed_material, seedlen).
3. V = seed.
4. C = Hash_df ((0x00 || V), seedlen).
5. reseed_counter = 1.
6. Return (V, C, reseed_counter).

HASH


(Opt.)
additional
input
V
C
reseed
counter
0x02 || V || additional input
Hash Function
V
+
If additional input ≠ Null
C
ctr
+
+
V
+
1
+
Hash Function
0x03 || V
Iterate to obtain enough bits
+
Counter (From 1)
Hash Function
Pseudorandom Bits
V
reseed counter
C

Ďakujem za pozornosť ☺

http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-90Ar1.pdf =zdroj