Random Thoughts: Bringing Ephemerality to the Deterministic World

Random Thoughts: Bringing Ephemerality to the Deterministic World

"Random Number Generation is too important to be left to chance"
Robert Coveyou, American mathematician

Ephemeral Event

Randomization used in an encryption scheme

Provably Independent Event: Random yes, Ephemeral no

Integer Agreement

Ephemeral Event

Randomization used in an encryption scheme

RSA

$$p, q \in_R prime$$
$$n = p * q$$
$$e * d \equiv 1 \bmod \phi(n)$$

$$c = m^e \bmod n$$
$$m = c^d \bmod n$$

ElGamal

$$g \in \mathcal{G}_q$$
$$s \in_R \mathbb{Z}_q$$
$$h = g^s \bmod p$$

$$r \in_R \mathbb{Z}_q$$
$$c = (g^r, h^r * m) \bmod p$$
$$m = (h^r * m) * (g^r)^{-s} \bmod p$$
$$m = (h^r * m) * h^{-r} \bmod p$$

Provably Independent Event: Random yes, Ephemeral no

Integer Agreement

Common Parameters
g, q, p

Alice:
$\alpha \in \mathbb{Z}_q$
Commitment to α
$\mapsto g^\alpha \bmod p$

After having seen Bobs commitment
$\mapsto \alpha$

Bob:
$\beta \in \mathbb{Z}_q$
Commitment to β
$\mapsto g^\beta \bmod p$

After having seen Alices commitment
$\mapsto \beta$

Integer Agreement

Common Parameters

g, q, p

Alice:

$\alpha \in \mathbb{Z}_q$

Commitment to α

$\mapsto g^\alpha \bmod p$

After having seen Bobs commitment

$\mapsto \alpha$

Bob:

$\beta \in \mathbb{Z}_q$

Commitment to β

$\mapsto g^\beta \bmod p$

After having seen Alices commitment

$\mapsto \beta$

$$\gamma = \alpha \oplus \beta$$

Alice:
$\alpha \in \mathbb{Z}_q$
Commitment to α
$\mapsto g^\alpha \bmod p$

After having seen Bobs commitment
$\mapsto \alpha$

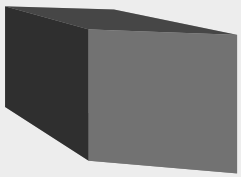
Bob:
$\beta \in \mathbb{Z}_q$
Commitment to β
$\mapsto g^\beta \bmod p$

After having seen Alices commitment
$\mapsto \beta$

$$\gamma = \alpha \oplus \beta$$

Alice and Bob know that they have chosen their values independent and hence the result is independent... but they cannot prove that to anyone else.
What, if they collude?

Random Oracle Model



$\mathrm{RandomOracle}: (0,1)^+ \mapsto (0,1)^n$

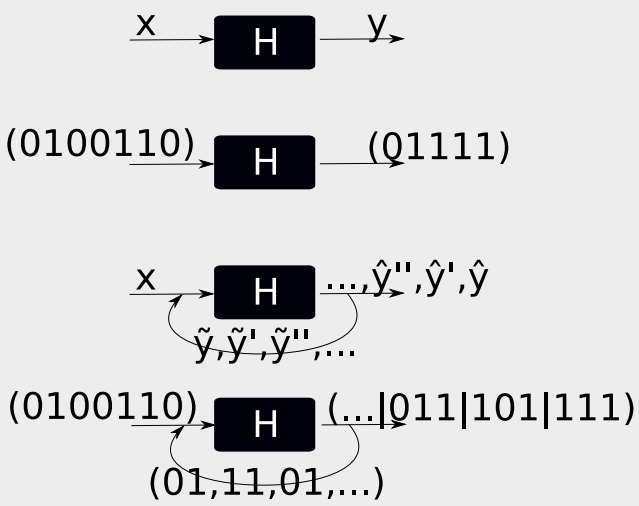
Properties
- Smallest function mapping each possible query to a (fixed) random response from its output domain.
- No dependency amongst different random responses
- Each random response drawn uniformly at random

Implication
- No response is predictable (Highest level of surprise)
- $RandomOracle(x) \mapsto y$ allways (No surprise)

= *It is a function, though fully deterministic*

Usage
- Make Random Oracle universally known
- Universally decide on an input string

Implementation
- Cryptographic Hash-Function $H(x) \mapsto$ Single value
- Looping $H(x) \mapsto$ Pseudo Random Stream



setup

refresh

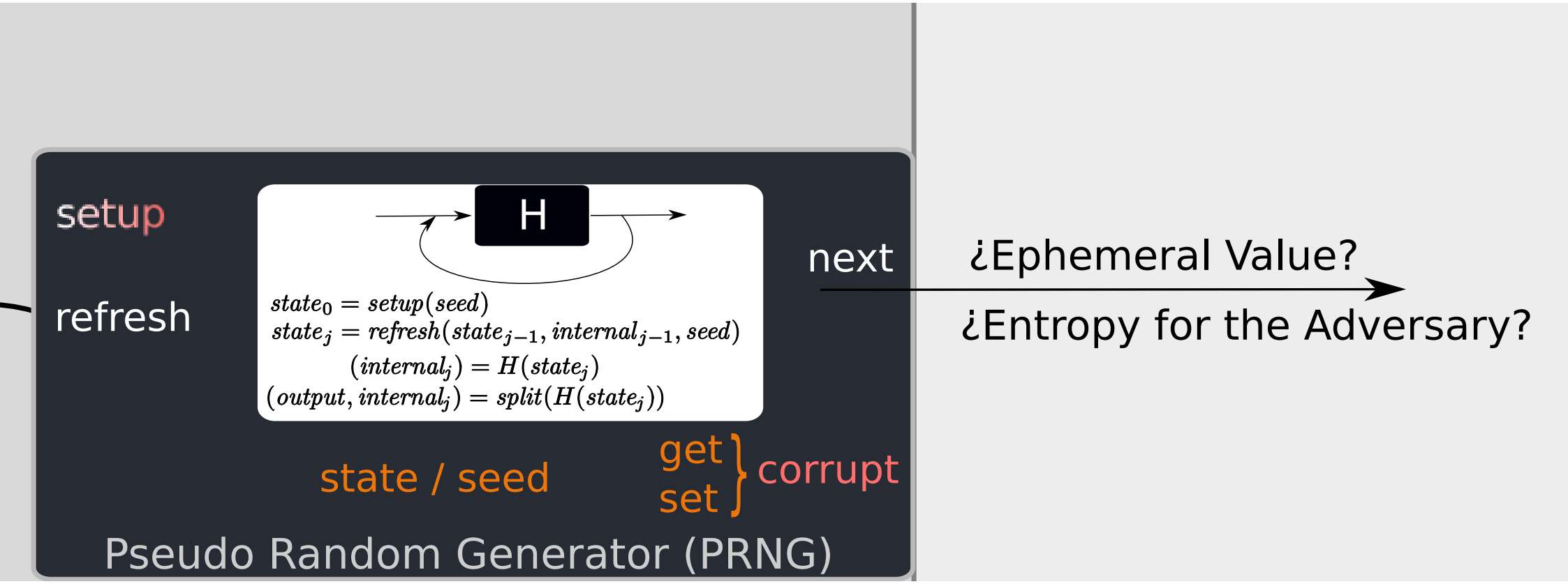
$$state_0 = setup(seed)$$
$$state_j = refresh(state_{j-1}, internal_{j-1}, seed)$$
$$(internal_j) = H(state_j)$$
$$(output, internal_j) = split(H(state_j))$$

next

state / seed

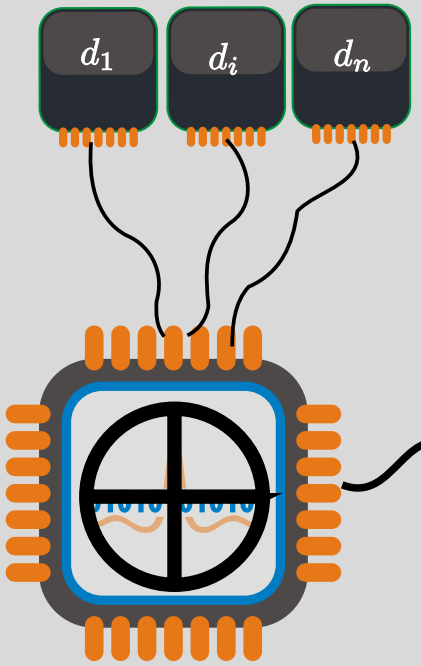
get
set
corrupt

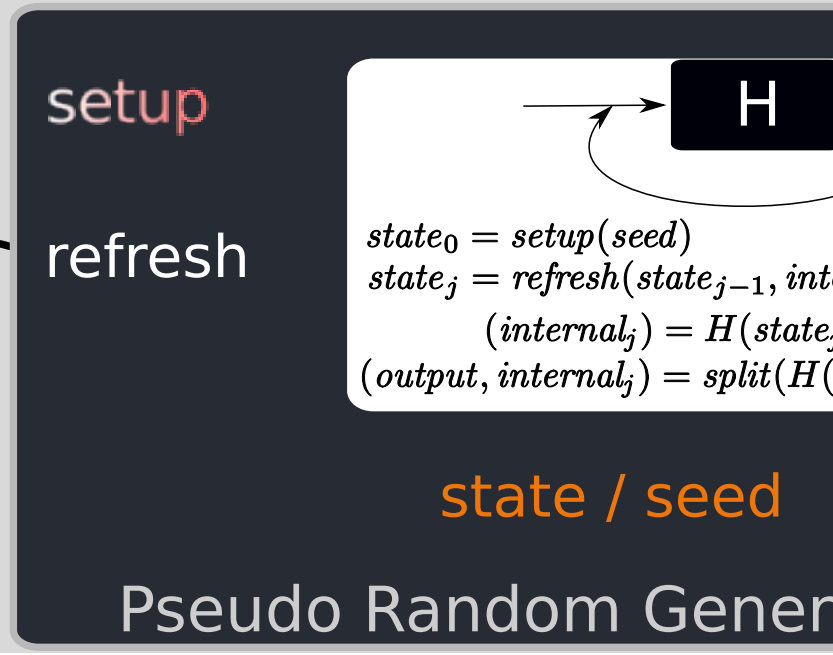
Pseudo Random Generator (PRNG)

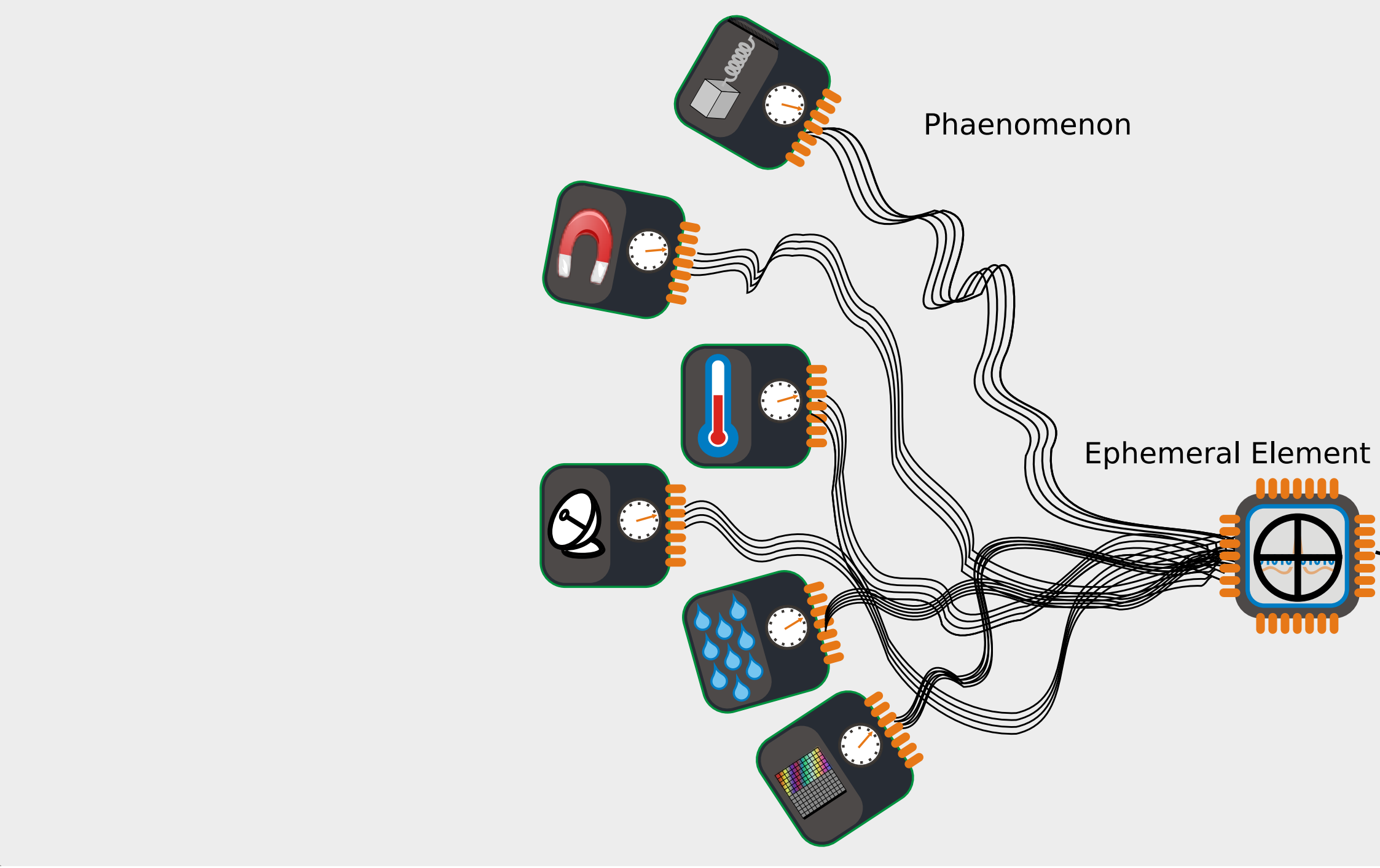

setup
refresh
H
$state_0 = setup(seed)$
$state_j = refresh(state_{j-1}, internal_{j-1}, seed)$
$(internal_j) = H(state_j)$
$(output, internal_j) = split(H(state_j))$
state / seed
get
set
corrupt
Pseudo Random Generator (PRNG)
next
¿Ephemeral Value?
¿Entropy for the Adversary?

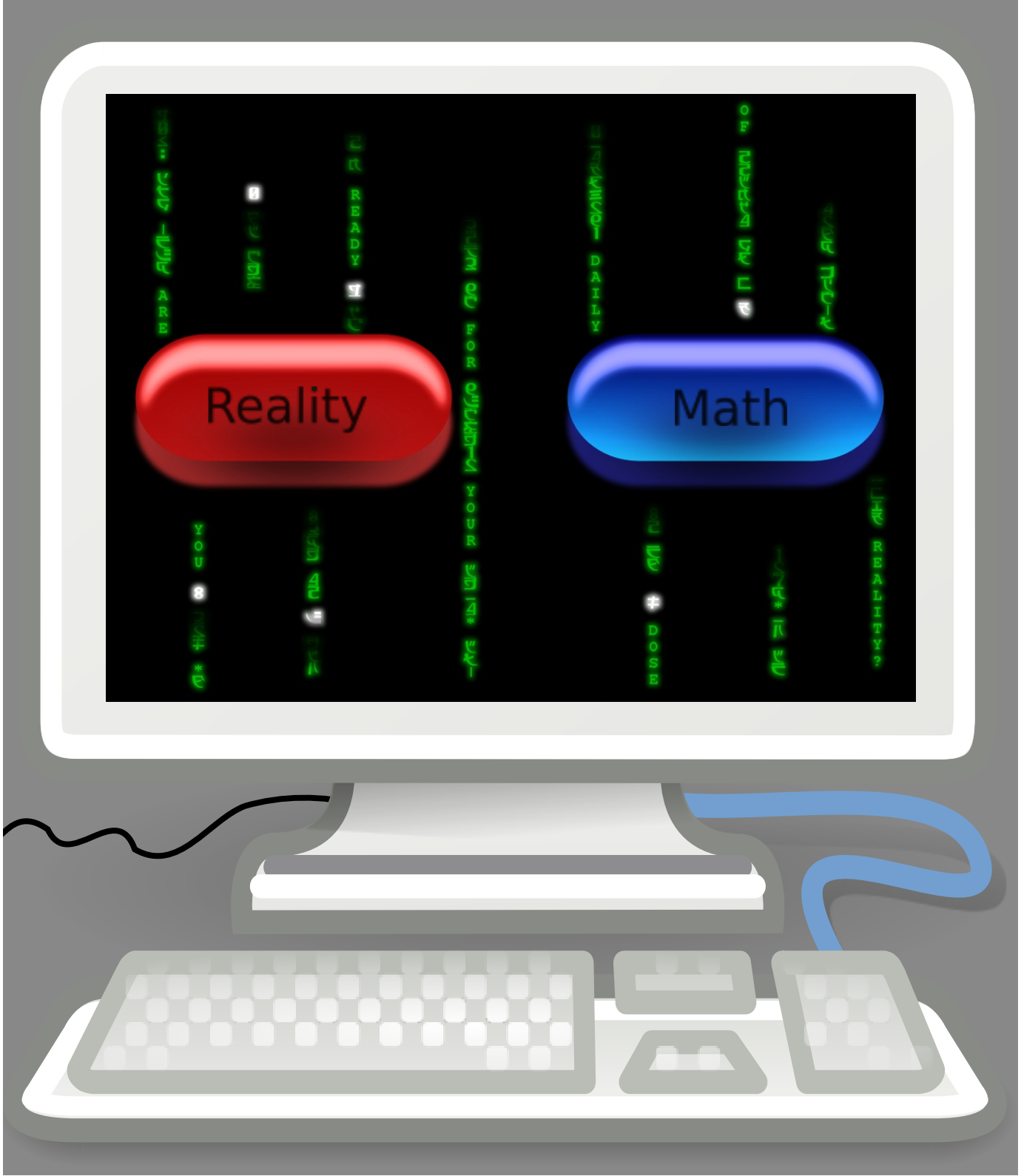
$\mathcal{D}$ =Distribution Sampler
d_i = Distribution
$d_{i,j}$ = Distribution Sample
$D = d_1, \ldots, d_n$
s = Security Parameter
$seed = \mathcal{D}(D, s)$

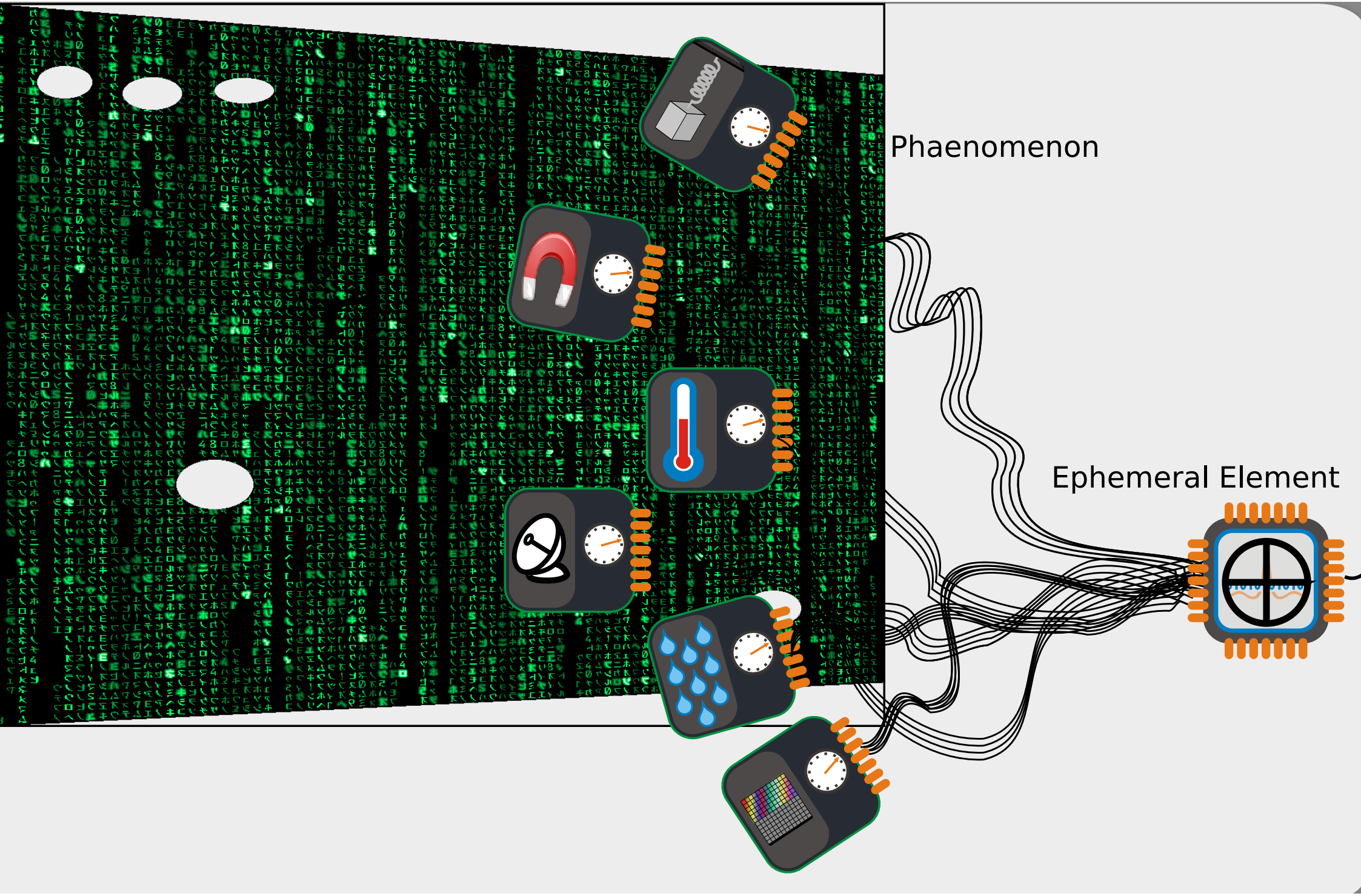
$$seed = (d_{1,1} \oplus d_{i,1} \oplus d_{n,1}, \ldots, d_{1,s} \oplus d_{i,s} \oplus d_{n,s})$$



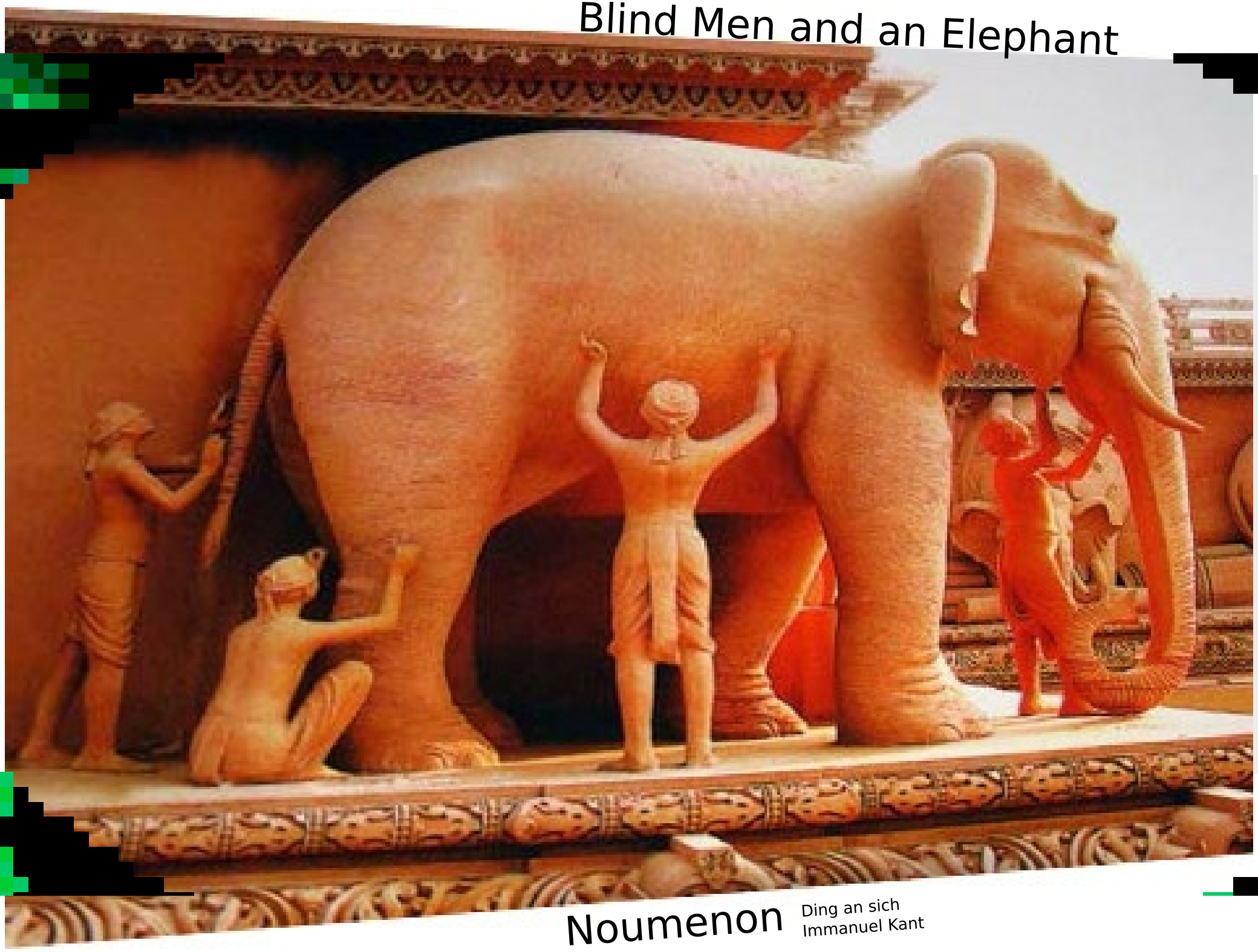



Phaenomenon
Ephemeral Element


Reality
Math
ARE YOU READY FOR YOUR DAILY DOSE OF REALITY?


Phaenomenon
Ephemeral Element

Blind Men and an Elephant



Noumenon

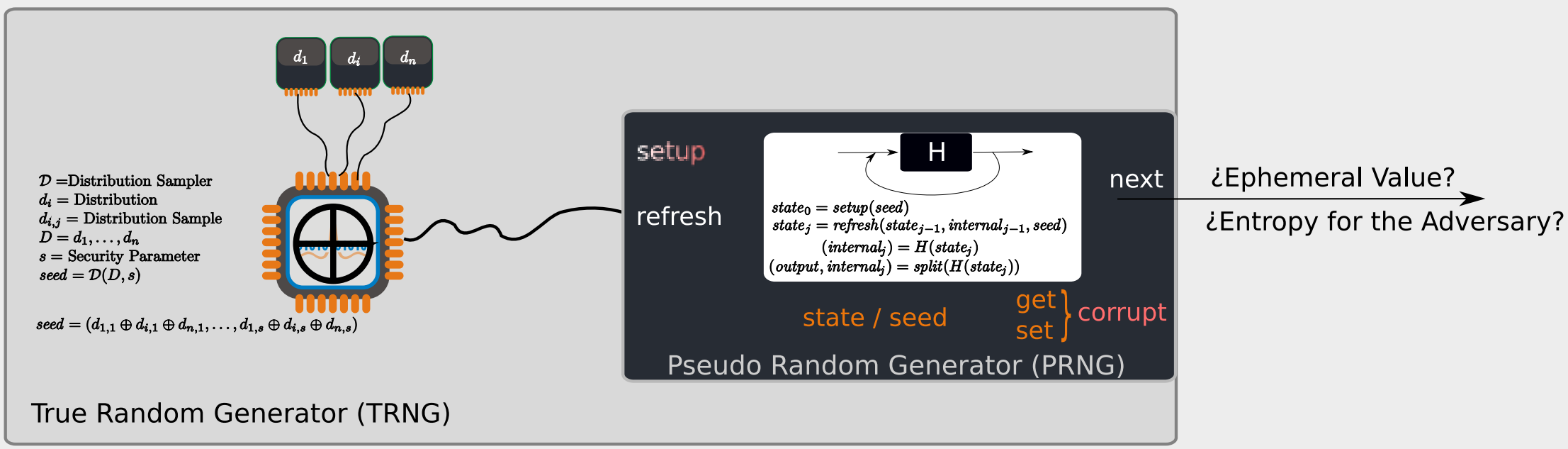
Ding an sich
Immanuel Kant

Blind Men and an Elephant

Noumenon

Ding an sich
Immanuel Kant


and an Elephant
NATIONAL SECURITY AGENCY
UNITED STATES OF AMERICA
Phaenomenon
Ephemeral
Ding an sich
Immanuel Kant


d_1
d_i
d_n
$\mathcal{D}$ =Distribution Sampler
d_i = Distribution
$d_{i,j}$ = Distribution Sample
$D = d_1, \ldots, d_n$
s = Security Parameter
$seed = \mathcal{D}(D, s)$
$seed = (d_{1,1} \oplus d_{i,1} \oplus d_{n,1}, \ldots, d_{1,s} \oplus d_{i,s} \oplus d_{n,s})$
True Random Generator (TRNG)
setup
refresh
H
$state_0 = setup(seed)$
$state_j = refresh(state_{j-1}, internal_{j-1}, seed)$
$(internal_j) = H(state_j)$
$(output, internal_j) = split(H(state_j))$
state / seed
get
set
corrupt
Pseudo Random Generator (PRNG)
next
¿Ephemeral Value?
¿Entropy for the Adversary?

“Random Number Generation is too important to be left to chance”
Robert Coveyou, American mathematician

Random Thoughts: Bringing Ephemerality to the Deterministic World

"Random Number Generation is too important to be left to chance"
Robert Coveyou, American mathematician

Ephemeral Event

Randomization used in an encryption scheme

RSA

$p, q \in_R prime$
$n = p * q$
$e * d \equiv 1 \bmod \phi(n)$

$c = m^e \bmod n$
$m = c^d \bmod n$

ElGamal

$g \in \mathcal{G}_q$
$s \in_R \mathbb{Z}_q$
$h = g^s \bmod p$

$r \in_R \mathbb{Z}_q$
$c = (g^r, h^r * m) \bmod p$
$m = (h^r * m) * (g^r)^{-s} \bmod p$
$m = (h^r * m) * h^{-r} \bmod p$

http://www.tu.no/it/2013/09/05/feil-krypteringer-avslo-stemmer

Provably Independent Event: Random yes, Ephemeral no

Integer Agreement

Common Parameters
g, q, p

Alice:
$\alpha \in \mathbb{Z}_q$
Commitment to α
$\mapsto g^\alpha \bmod p$

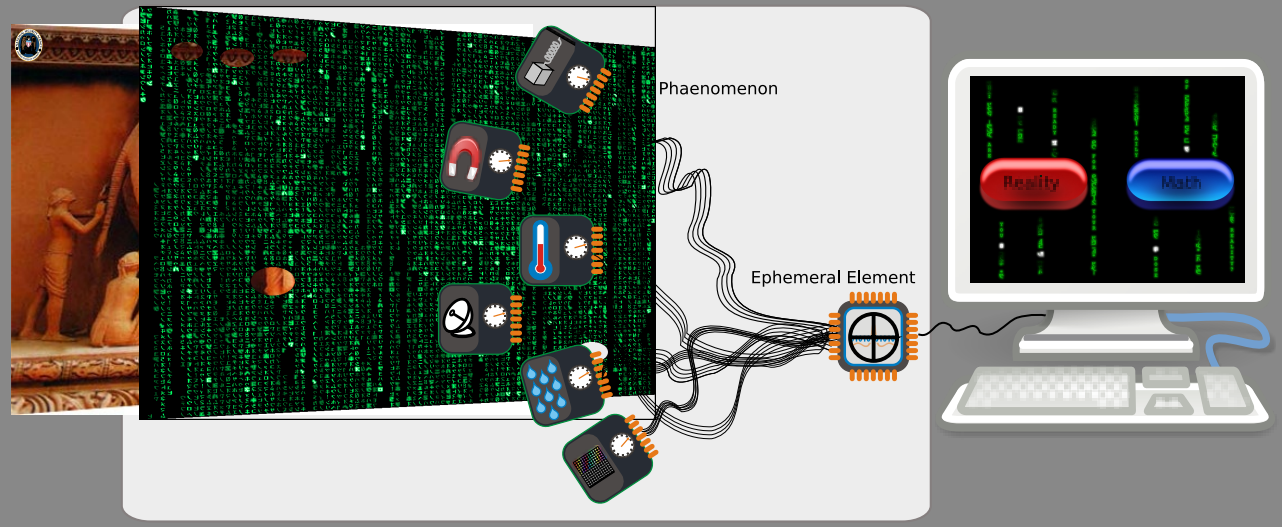
After having seen Bobs commitment
$\mapsto \alpha$

Bob:
$\beta \in \mathbb{Z}_q$
Commitment to β
$\mapsto g^\beta \bmod p$

After having seen Alices commitment
$\mapsto \beta$

$\gamma = \alpha \oplus \beta$

Alice and Bob know that they have chosen their values independent and hence the result is independent...
but they cannot prove that to anyone else.
What, if they collude?



Random Oracle Model

$\text{RandomOracle}: (0,1)^+ \mapsto (0,1)^n$

Properties
- Smallest function mapping each possible query to a (fixed) random response from its output domain.
- No dependency amongst different random responses
- Each random response drawn uniformly at random

Implication
- No response is predictable (Highest level of surprise)
- $RandomOracle(x) \mapsto y$ allways (No surprise)

= *It is a function, though fully deterministic*

Usage
- Make Random Oracle universally known
- Universally decide on an input string

Implementation
- Cryptographic Hash-Function $H(x) \mapsto$ Single value
- Looping $H(x) \mapsto$ Pseudo Random Stream