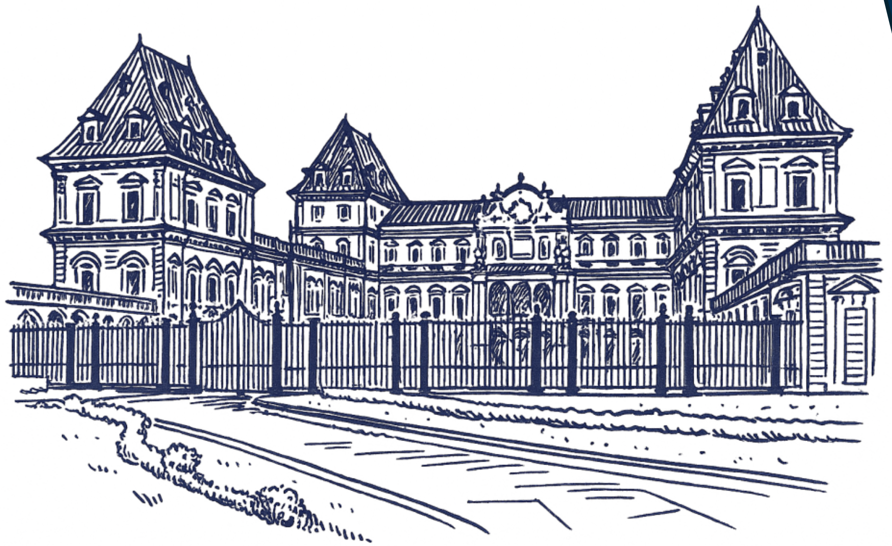


Cryptographic Standards for RNGs

Leonardo Errati, Roberto La Scala, Mauro Patano

Cifris25 – Sept. 12th, 2025

Politecnico di Torino

UNIVERSITÀ DEGLI STUDI DI BARI ALDO MORO

INFN

Why randomness?

$$x \xleftarrow{\$} S$$

«Fully Adaptive Schnorr Threshold Signatures»
Crites, Komlo, Maller

$\mathsf{Setup}(1^\kappa)$

$(\mathbb{G}, p, g) \leftarrow \mathsf{GrGen}(1^\kappa)$

// select two hash functions

$\mathsf{H}_{\mathsf{cm}}, \mathsf{H}_{\mathsf{sig}} : \{0,1\}^* \rightarrow \mathbb{Z}_p$

$\mathsf{par}_{\mathsf{DS}} \leftarrow \mathsf{DS.Setup}(1^\kappa)$

$\mathsf{par} \leftarrow ((\mathbb{G}, p, g), \mathsf{H}_{\mathsf{cm}}, \mathsf{H}_{\mathsf{sig}}, \mathsf{par}_{\mathsf{DS}})$

return par

$\mathsf{KeyGen}(n, t+1)$

$x \leftarrow_{\$} \mathbb{Z}_p$; $\mathsf{pk} \leftarrow g^x$

$\{(i, x_i)\}_{i \in [n]} \leftarrow_{\$} \mathsf{IssueShares}(x, n, t+1)$

// Shamir secret sharing of x

for $i \in [n]$ **do**

$X_i \leftarrow g^{x_i}$; $(\hat{X}_i, \hat{x}_i) \leftarrow_{\$} \mathsf{DS.KeyGen}()$

$\mathsf{pk}_i \leftarrow (X_i, \hat{X}_i)$

$\mathsf{sk}_i \leftarrow (x_i, \hat{x}_i, \mathsf{pk}, \{\mathsf{pk}_i\}_{i \in [n]})$

return $(\mathsf{pk}, \{(\mathsf{pk}_i, \mathsf{sk}_i)\}_{i \in [n]})$

$\mathsf{Sign}_3(k, \mathsf{st}_{k,2}, \{(R_i, \hat{\sigma}_i)\}_{i \in \mathcal{S}})$

parse

$(R'_k, r_k, \mathsf{cm}_k, \hat{\sigma}'_k, \mathcal{S}', m, \{\mathsf{cm}_i\}_{i \in \mathcal{S}'}, \mathsf{sk}_k)$

$\leftarrow \mathsf{st}_{k,2}$

return $\perp$ **if**

$(R'_k, \hat{\sigma}'_k, \mathcal{S}') \neq (R_k, \hat{\sigma}_k, \mathcal{S})$

// checks inputs against records in state

for $i \in \mathcal{S}$ **do**

return $\perp$ **if** $\mathsf{cm}_i \neq \mathsf{H}_{\mathsf{cm}}(i, R_i)$

$\mathsf{msg}_i \leftarrow (i, \mathsf{cm}_i, R_i, \mathcal{S}, m, \{\mathsf{cm}_j\}_{j \in \mathcal{S}})$

if $\mathsf{DS.Verify}(\hat{X}_i, \mathsf{msg}_i, \hat{\sigma}_i) \neq 1$

return $\perp$

$R \leftarrow \prod_{i \in \mathcal{S}} R_i$

$c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{pk}, m, R)$

$z_k \leftarrow r_k + c\lambda_k x_k$

// λ_k is the Lagrange coefficient for k

return z_k

«Fully Adaptive Schnorr Threshold Signatures»
Crites, Komlo, Maller

$\mathsf{Setup}(1^\kappa)$

$\mathsf{ar}_{\mathsf{DS}})$

$\mathsf{s}(x, n, t+1)$

x

$\mathsf{S.KeyGen}()$

$_{[n]})$

$_{i \in [n]})$

$\mathsf{Sign}_3(k, \mathsf{st}_{k,2}, \{(R_i, \hat{\sigma}_i)\}_{i \in \mathcal{S}})$

parse $(R'_k, r_k, \mathsf{cm}_k, \hat{\sigma}'_k, \mathcal{S}', m, \{\mathsf{cm}_i\}_{i \in \mathcal{S}'}, \mathsf{sk}_k) \leftarrow \mathsf{st}_{k,2}$

return $\perp$ **if** $(R'_k, \hat{\sigma}'_k, \mathcal{S}') \neq (R_k, \hat{\sigma}_k, \mathcal{S})$

// checks inputs against records in state

for $i \in \mathcal{S}$ **do**

  return $\perp$ **if** $\mathsf{cm}_i \neq \mathsf{H}_{\mathsf{cm}}(i, R_i)$

  $\mathsf{msg}_i \leftarrow (i, \mathsf{cm}_i, R_i, \mathcal{S}, m, \{\mathsf{cm}_j\}_{j \in \mathcal{S}})$

  if $\mathsf{DS.Verify}(\hat{X}_i, \mathsf{msg}_i, \hat{\sigma}_i) \neq 1$

    return $\perp$

$R \leftarrow \prod_{i \in \mathcal{S}} R_i$

$c \leftarrow \mathsf{H}_{\mathsf{sig}}(\mathsf{pk}, m, R)$

$z_k \leftarrow r_k + c\lambda_k x_k$

// λ_k is the Lagrange coefficient for k

return z_k

Public Data	System parameters $q, n, k \in \mathbb{N}$, hash function HASH, canonical form function CF.
Private Key	$\mu \in M_n$.
Public Key	$(\mathbf{G}, \mathbf{G}')$ with $\mathbf{G} \xleftarrow{\$} \mathbb{F}_q^{k \times n}$ and $\mathbf{G}' = \mathrm{RREF}(\mu(\mathbf{G}))$.

PROVER | VERIFIER

do:

  $\widetilde{\mu} \xleftarrow{\$} M_n$

  $\big((\mathbf{I}_k \mid \mathbf{A}), \pi\big) \leftarrow \mathrm{RREF}^*(\widetilde{\mu}(\mathbf{G}))$

while $\mathrm{CF}(\mathbf{A}) = \perp$

$\mathsf{cmt} \leftarrow \mathrm{HASH}(\mathrm{CF}(\mathbf{A}))$ $\xrightarrow{\mathsf{cmt}}$

$\xleftarrow{\mathsf{ch}}$ $\mathsf{ch} \xleftarrow{\$} \{0, 1\}$

If $\mathsf{ch} = 0$:

  $\mathsf{rsp} \leftarrow \widetilde{\mu}$

Else:

  $\mathsf{rsp} \leftarrow J = \mathrm{COSETREP}(\pi \circ \widetilde{\mu} \circ \mu^{-1})$ $\xrightarrow{\mathsf{rsp}}$

VERIFIER:

If $\mathsf{ch} = 0$:

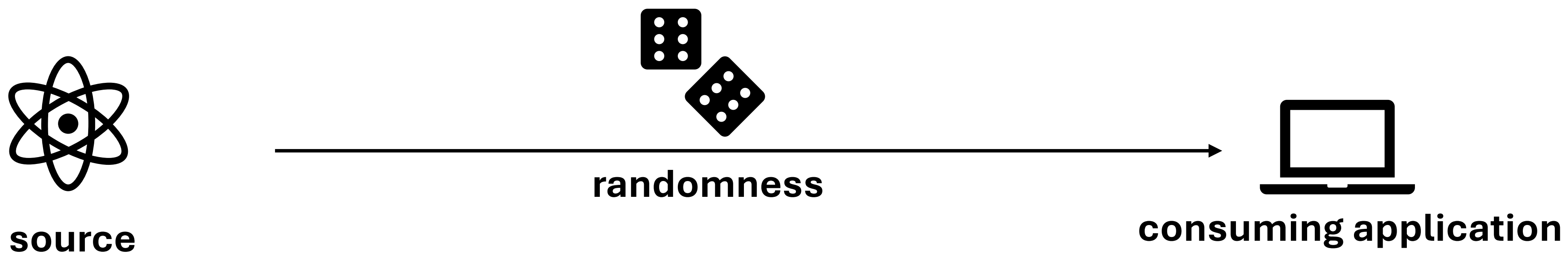
  $(\mathbf{I}_k \mid \mathbf{A}) \leftarrow \mathrm{RREF}^*(\mathsf{rsp}(\mathbf{G}))$

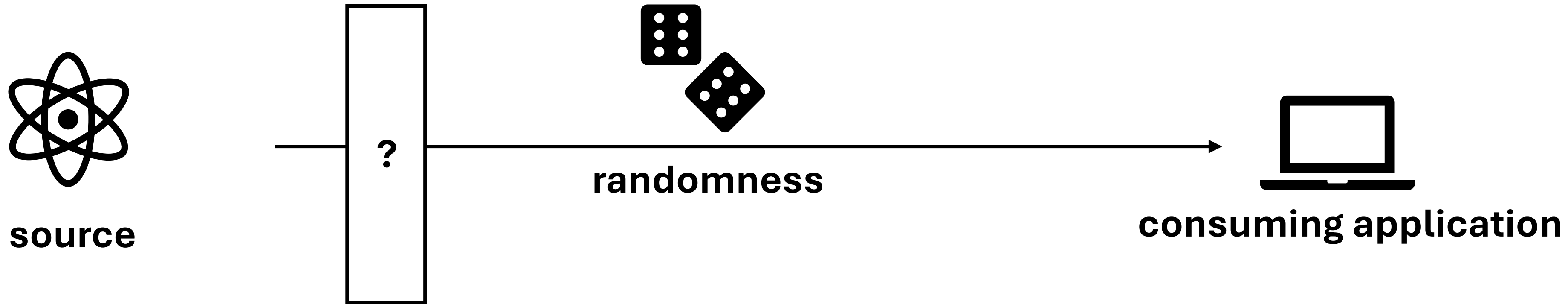
Else:

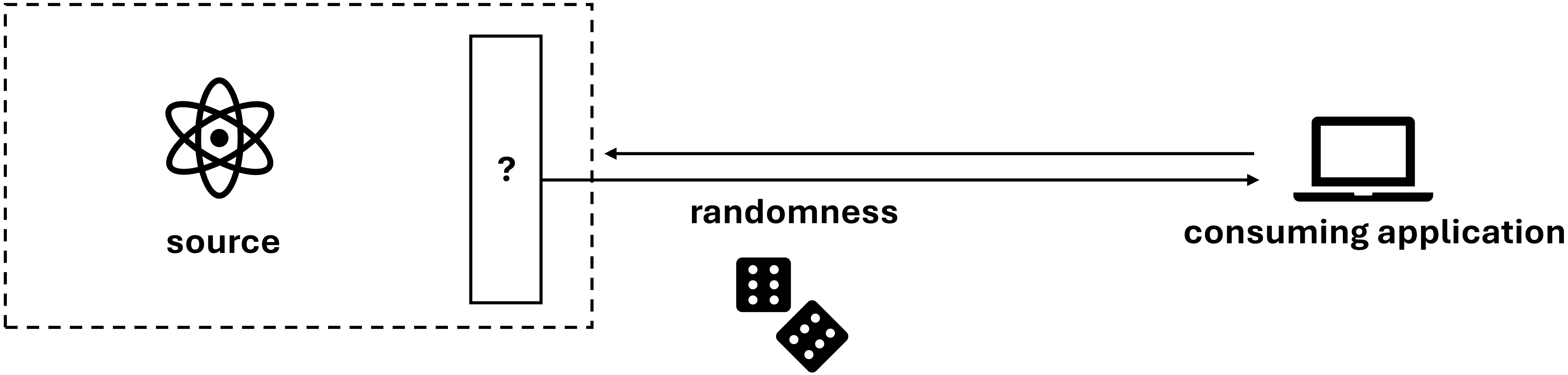
  $\mathbf{A} \leftarrow \mathbf{G}'^{-1}_{\mathsf{rsp}} \cdot \mathbf{G}'_{\{1,\cdots,n\} \setminus \mathsf{rsp}}$

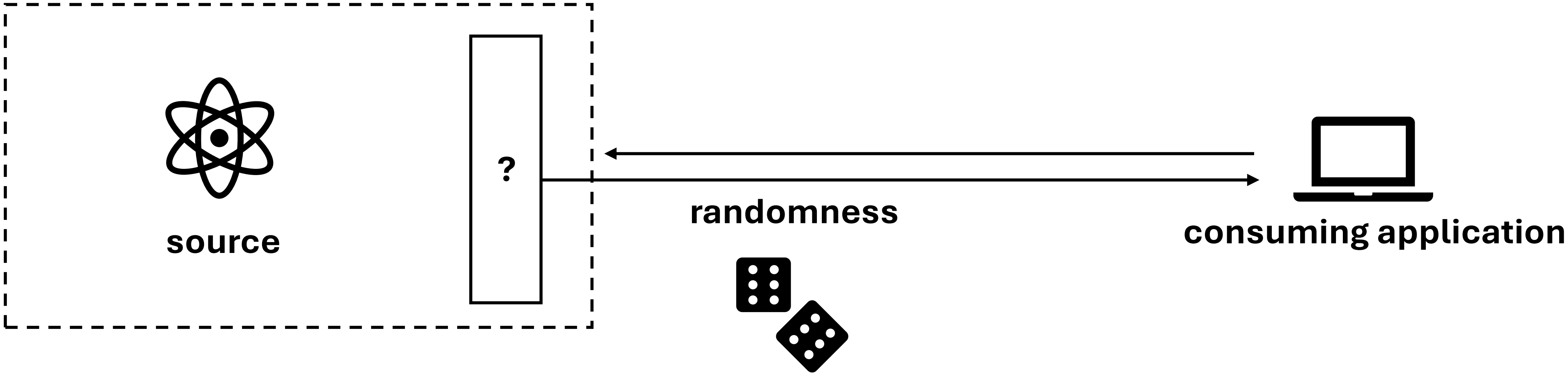
Accept if $\mathrm{HASH}(\mathrm{CF}(\mathbf{A})) = \mathsf{cmt}$

«LESS : Linear Equivalence Signature Scheme»
Baldi et al.


randomness
source
consuming application

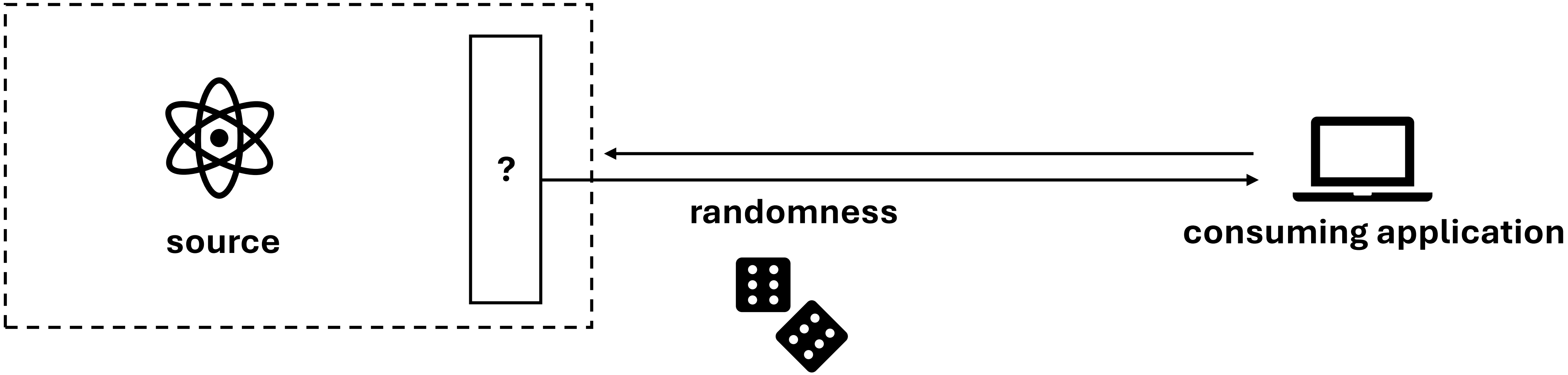

?
randomness
source
consuming application


?
source
randomness
consuming application



RBG: A device or algorithm that outputs a sequence of binary bits that appears to be statistically independent and unbiased.

Non-deterministic RBG: always has access to fresh entropy, its output bitstrings that have full entropy.



NIST Special Publication 800-90A
Revision 1

Recommendation for Random Number Generation Using Deterministic Random Bit Generators

NIST Special Publication 800-90B

Recommendation for the Entropy Sources Used for Random Bit Generation

NIST Special Publication 800
NIST SP 800-90C 4pd

Recommendation for Random Bit Generator (RBG) Constructions

Fourth Public Draft

The NIST SP 800-90 framework

SP 800-90A: DRBG constructions

SP 800-90A: DRBG constructions

Finite-state machine with interfaces:

- instantiate / uninstantiate
- generate
- reseed

SP 800-90A: DRBG constructions

Finite-state machine with interfaces:

- instantiate / uninstantiate
- generate
- reseed

DRBGs are based on cryptographic primitives:

- HMAC
- Hash functions
- CTR-mode block ciphers
- Dual_EC_DRGB

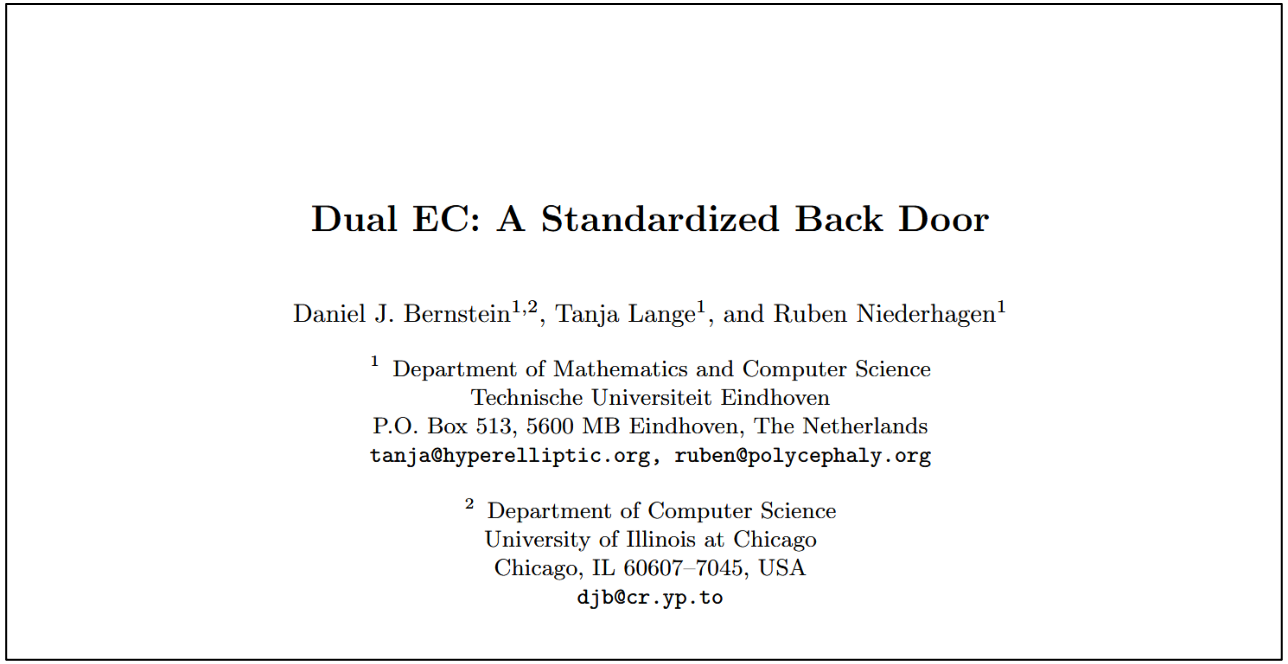
SP 800-90A: DRBG constructions

Finite-state machine with interfaces:

- instantiate / uninstantiate
- generate
- reseed

DRBGs are based on cryptographic primitives:

- HMAC
- Hash functions
- CTR-mode block ciphers
- ~~Dual_EC_DRGB~~



Dual EC: A Standardized Back Door

Daniel J. Bernstein[1,2], Tanja Lange[1], and Ruben Niederhagen[1]

[1] Department of Mathematics and Computer Science
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven, The Netherlands
tanja@hyperelliptic.org, ruben@polycephaly.org

[2] Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607–7045, USA
djb@cr.yp.to

SP 800-90A: DRBG constructions

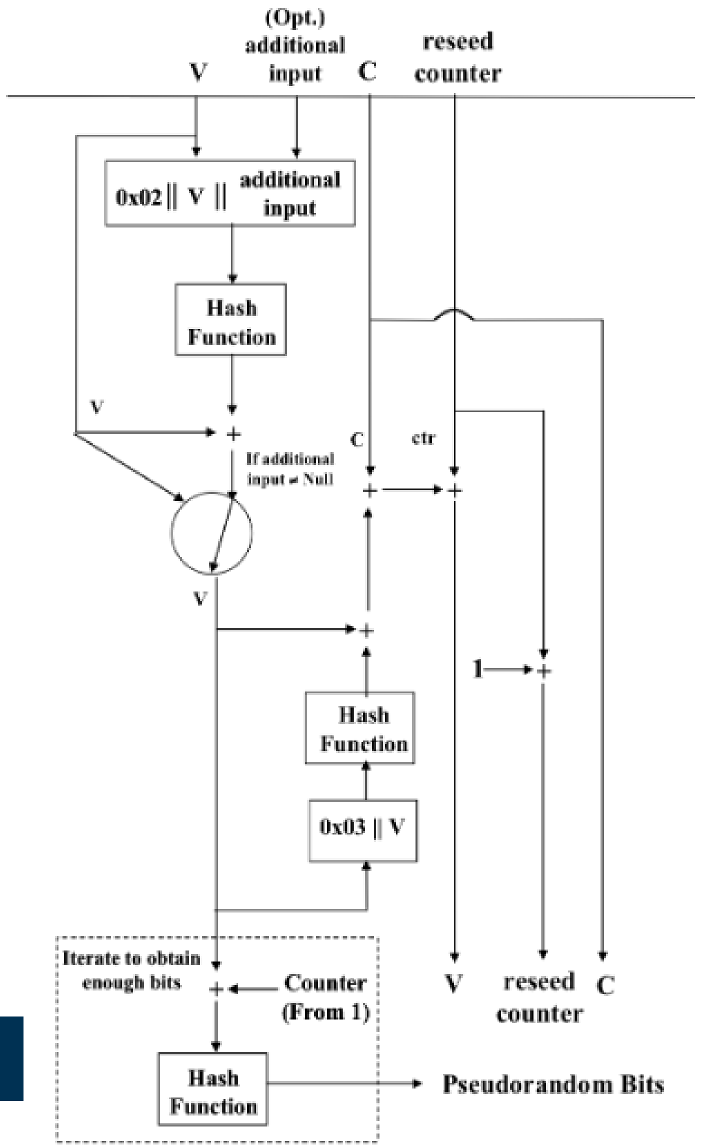
Finite-state machine with interfaces:

- instantiate / uninstantiate
- generate
- reseed

DRBGs are based on cryptographic primitives:

- HMAC
- **Hash functions**
- CTR-mode block ciphers

HASH_DRBG



SP 800-90A: DRBG constructions

Finite-state machine with interfaces:

- instantiate / uninstantiate
- generate
- reseed

Security goals:

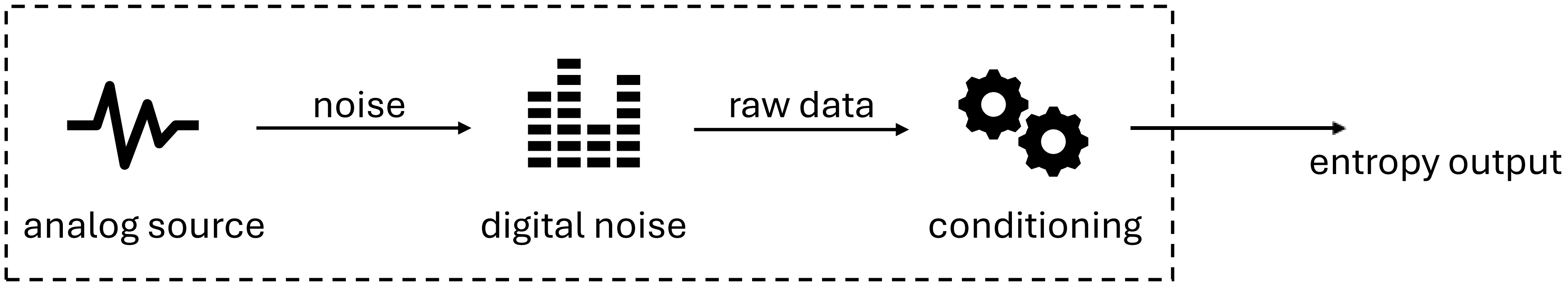
- backtracking resistance
- prediction resistance

DRBGs are based on cryptographic primitives:

- HMAC
- **Hash functions**
- CTR-mode block ciphers

SP 800-90B: entropy sources

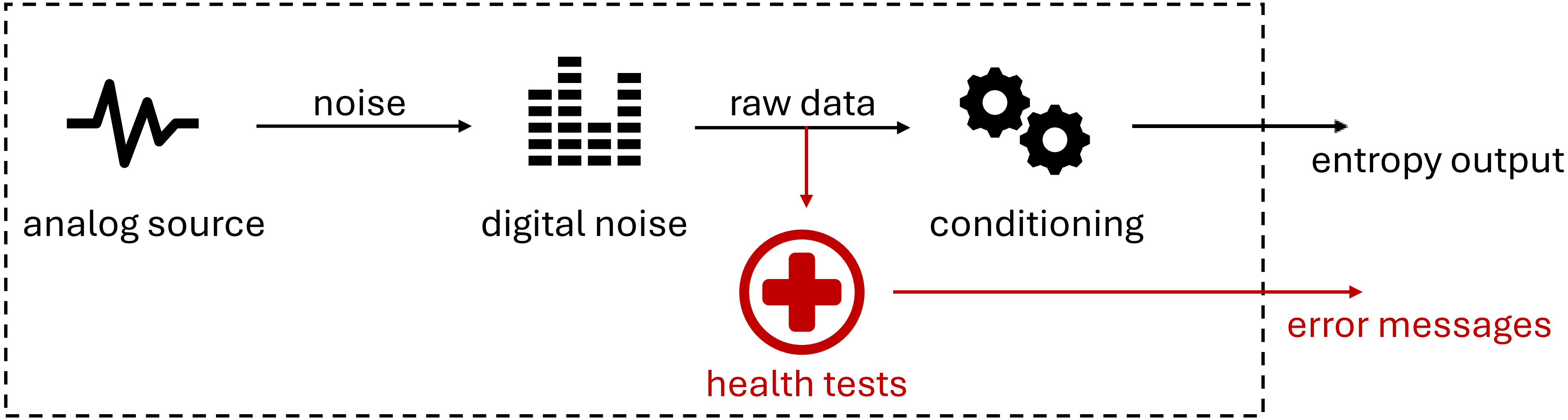
SP 800-90B: entropy sources



Noise source:

- physical / non-physical
- protected
- stationary distribution, ideally IID
- entropy estimate

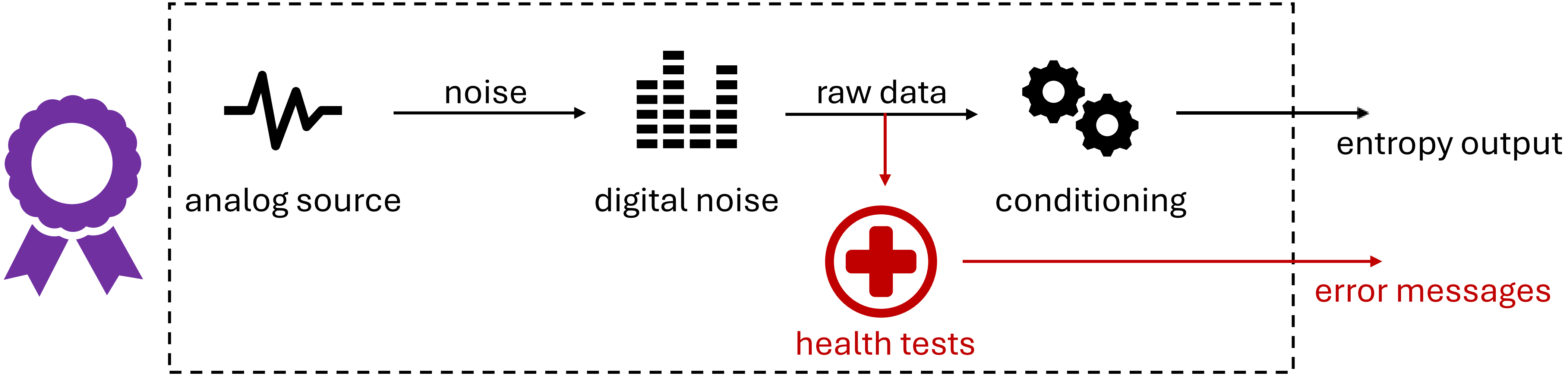
SP 800-90B: entropy sources



Noise source:
- physical / non-physical
- protected
- stationary distribution, ideally IID
- entropy estimate

Health tests:
- startup, restart, …
- continuous monitoring for catastrophic failures
- statistical tests

SP 800-90B: entropy sources



Noise source:
- physical / non-physical
- protected
- stationary distribution, ideally IID
- entropy estimate

Health tests:
- startup, restart, …
- continuous monitoring for catastrophic failures
- statistical tests

Must be **validated** by accredited laboratories.

SP 800-90C: combining constructions

SP 800-90C: combining constructions

Construction	Internal Entropy Source	Available randomness source for reseeding	Prediction Resistance	Full Entropy	Type of Randomness Source
RBG1	No	No	No	No	RBG2(P) or RBG3 construction
RBG2(P)	Yes	Yes	Optional	No	Physical entropy source
RBG2(NP)	Yes	Yes	Optional	No	Non-physical entropy source
RBG3(XOR) or RBG3(RS)	Yes	Yes	Yes	Yes	Physical entropy source
(Root) RBGC	Yes	Yes	Optional	No	RBG2 or RBG3 construction or Full-entropy source
(Non-root) RBGC	No	Yes	No	No	Parent RBGC construction

SP 800-90C: combining constructions

Construction	Internal Entropy Source	Available randomness source for reseeding	Prediction Resistance	Full Entropy	Type of Randomness Source
RBG1	No	No	No	No	RBG2(P) or RBG3 construction
RBG2(P)	Yes	Yes	Optional	No	Physical entropy source
RBG2(NP)	Yes	Yes	Optional	No	Non-physical entropy source
RBG3(XOR) or RBG3(RS)	Yes	Yes	Yes	Yes	Physical entropy source
(Root) RBGC	Yes	Yes	Optional	No	RBG2 or RBG3 construction or Full-entropy source
(Non-root) RBGC	No	Yes	No	No	Parent RBGC construction

SP 800-90C: combining constructions

Construction	Internal Entropy Source	Available randomness source for reseeding	Prediction Resistance	Full Entropy	Type of Randomness Source
RBG1	No	No	No	No	RBG2(P) or RBG3 construction
RBG2(P)	Yes	Yes	Optional	No	Physical entropy source
RBG2(NP)	Yes	Yes	Optional	No	Non-physical entropy source
RBG3(XOR) or RBG3(RS)	Yes	Yes	Yes	Yes	Physical entropy source
(Root) RBGC	Yes	Yes	Optional	No	RBG2 or RBG3 construction or Full-entropy source
(Non-root) RBGC	No	Yes	No	No	Parent RBGC construction

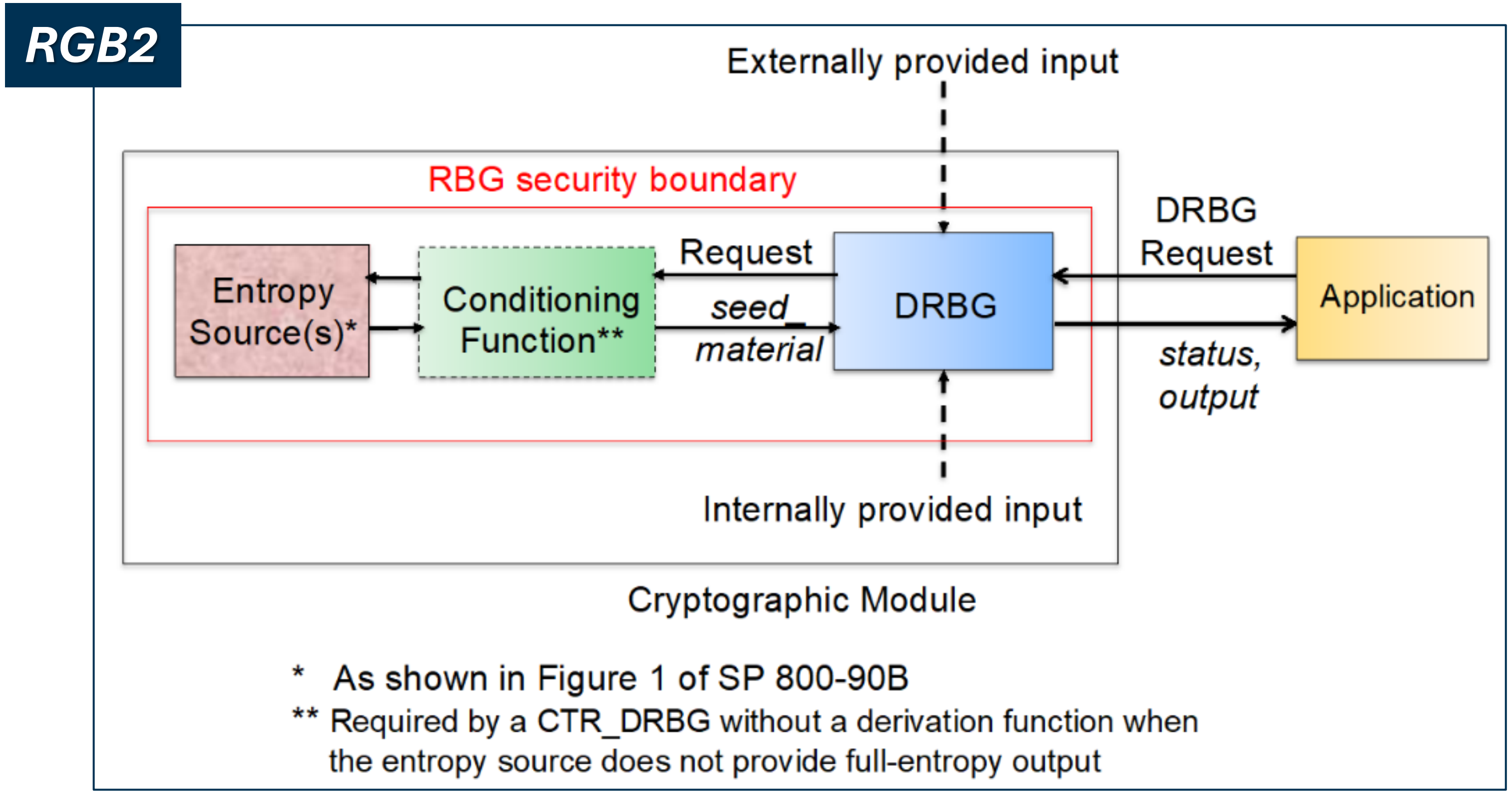
SP 800-90C: combining constructions

Construction	Internal Entropy Source	Available randomness source for reseeding	Prediction Resistance	Full Entropy	Type of Randomness Source
RBG1	No	No	No	No	RBG2(P) or RBG3 construction
RBG2(P)	Yes	Yes	Optional	No	Physical entropy source
RBG2(NP)	Yes	Yes	Optional	No	Non-physical entropy source
RBG3(XOR) or RBG3(RS)	Yes	Yes	Yes	Yes	Physical entropy source
(Root) RBGC	Yes	Yes	Optional	No	RBG2 or RBG3 construction or Full-entropy source
(Non-root) RBGC	No	Yes	No	No	Parent RBGC construction

SP 800-90C: combining constructions

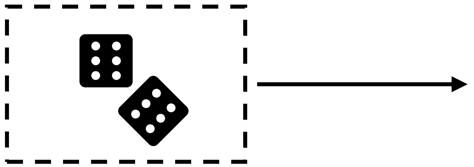
Construction	Internal Entropy Source	Available randomness source for reseeding	Prediction Resistance	Full Entropy	Type of Randomness Source
RBG1	No	No	No	No	RBG2(P) or RBG3 construction
RBG2(P)	Yes	Yes	Optional	No	Physical entropy source
RBG2(NP)	Yes	Yes	Optional	No	Non-physical entropy source
RBG3(XOR) or RBG3(RS)	Yes	Yes	Yes	Yes	Physical entropy source
(Root) RBGC	Yes	Yes	Optional	No	RBG2 or RBG3 construction or Full-entropy source
(Non-root) RBGC	No	Yes	No	No	Parent RBGC construction

SP 800-90C: combining constructions



Interlude: statistical tests

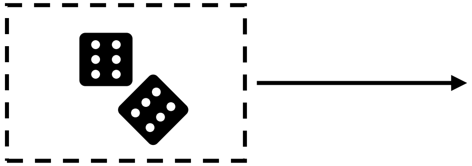
SP 800-22



10010111 01011011 01100111...

"TestU01: A C Library for Empirical Testing of Random Number Generators". L'Ecuyer and Simard.

SP 800-22



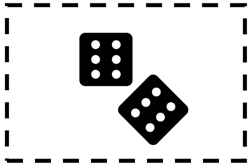
`10010111 01011011 01100111...`

Statistical tests:

$$\begin{cases} H_0\text{: the successive outputs } u_0, u_1, u_2, \dots, u_k \text{ are IID random variables } U\{0,1\} \\ H_1\text{: } H_0 \text{ is false} \end{cases}$$

"TestU01: A C Library for Empirical Testing of Random Number Generators". L'Ecuyer and Simard.

SP 800-22

 $\longrightarrow$ `10010111 01011011 01100111...` $\longrightarrow$ Y

Statistical tests:

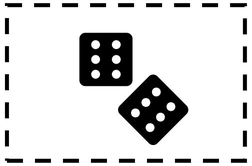
$$\begin{cases} H_0\text{: the successive outputs } u_0, u_1, u_2, \ldots, u_k \text{ are IID random variables } U\{0,1\} \\ H_1\text{: } H_0 \text{ is false} \end{cases}$$

We build a test statistics Y and study its distribution.

$$p = P[Y \geq y \mid H_0]$$

"TestU01: A C Library for Empirical Testing of Random Number Generators". L'Ecuyer and Simard.

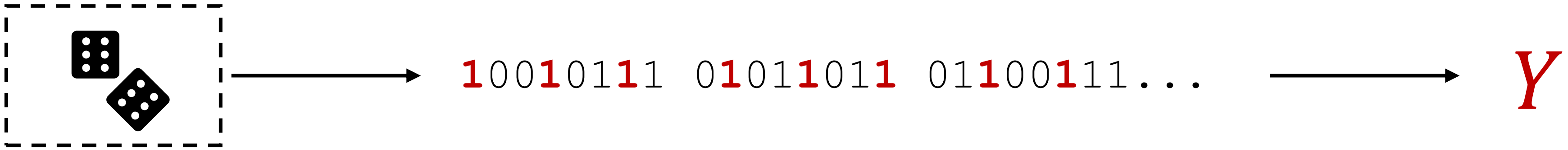
SP 800-22



10010111 01011011 01100111... → Y

“TestU01: A C Library for Empirical Testing of Random Number Generators”. L’Ecuyer and Simard.

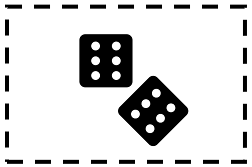
SP 800-22



NIST suite: 15 tests for different kinds of bias

“TestU01: A C Library for Empirical Testing of Random Number Generators”. L’Ecuyer and Simard.

SP 800-22



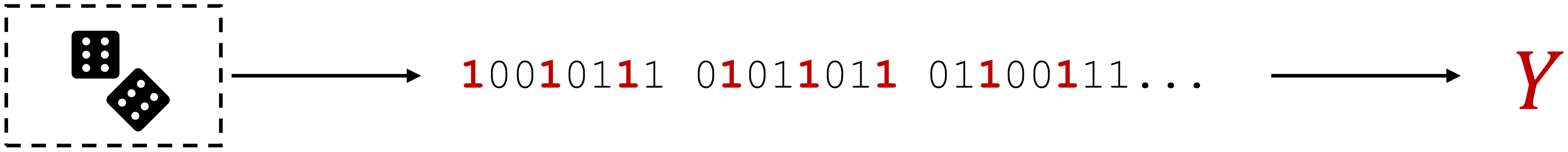
10010111 01011011 01100111... → Y

NIST suite: 15 tests for different kinds of bias

- **Frequency test**
  Under H_0, CLT-approximation of $Y = \sum_i u_i$

"TestU01: A C Library for Empirical Testing of Random Number Generators". L'Ecuyer and Simard.

SP 800-22



NIST suite: 15 tests for different kinds of bias

- **Frequency test**
  Under H_0, CLT-approximation of $Y = \sum_i u_i$
- **Random excursion test**
  Transform $u_0, u_1, u_2, \dots, u_k$ into a random walk on a graph
  Under H_0, the total visits to each state follow the discrete Markov distribution

"TestU01: A C Library for Empirical Testing of Random Number Generators". L'Ecuyer and Simard.

Randomness in the Wild

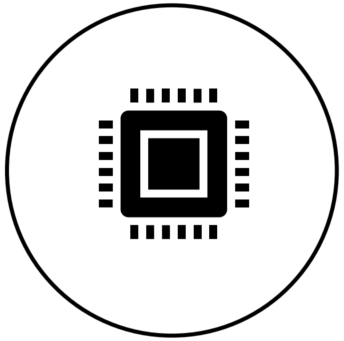
A joint effort

A joint effort

Physics:
- harness high-level entropy
- ensuring stability & avoiding bias

Cryptography:
- primitives for extraction & expansion
- security claims

Mathematics:
- statistical validation
- min-entropy estimation

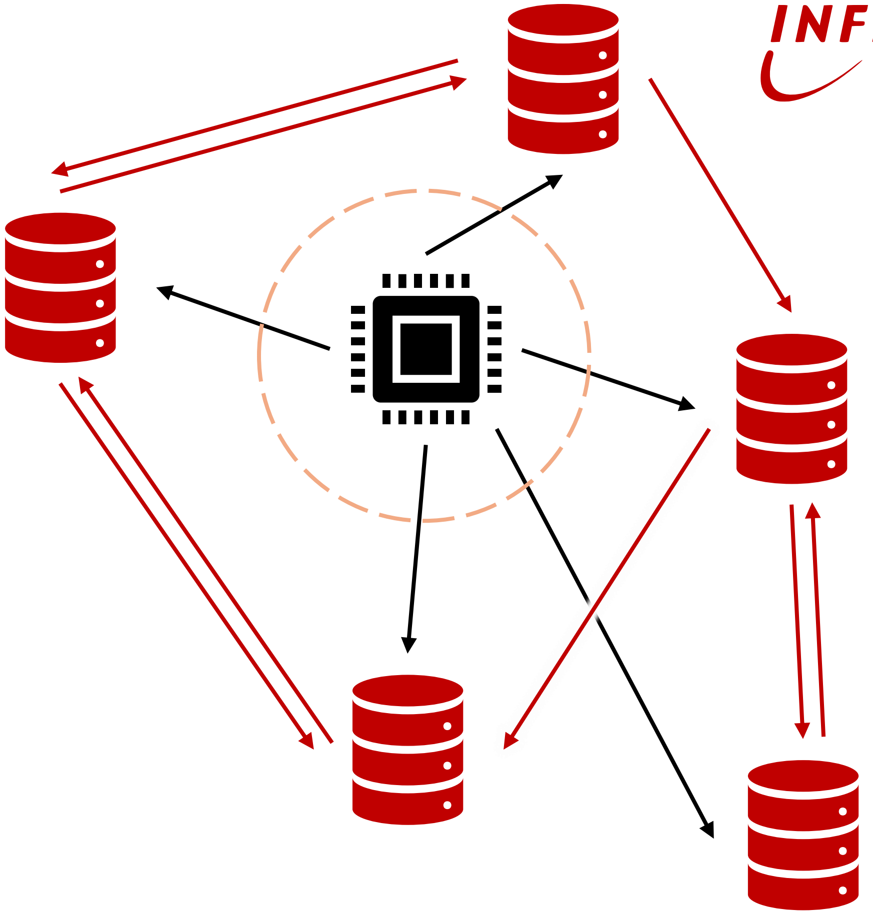


A joint effort (the case of INFN)

Physics:
- harness high-level quantum entropy
- ensuring stability & avoiding bias

Cryptography:
- primitives for extraction & expansion
- security claims

Mathematics:
- statistical validation
- min-entropy estimation



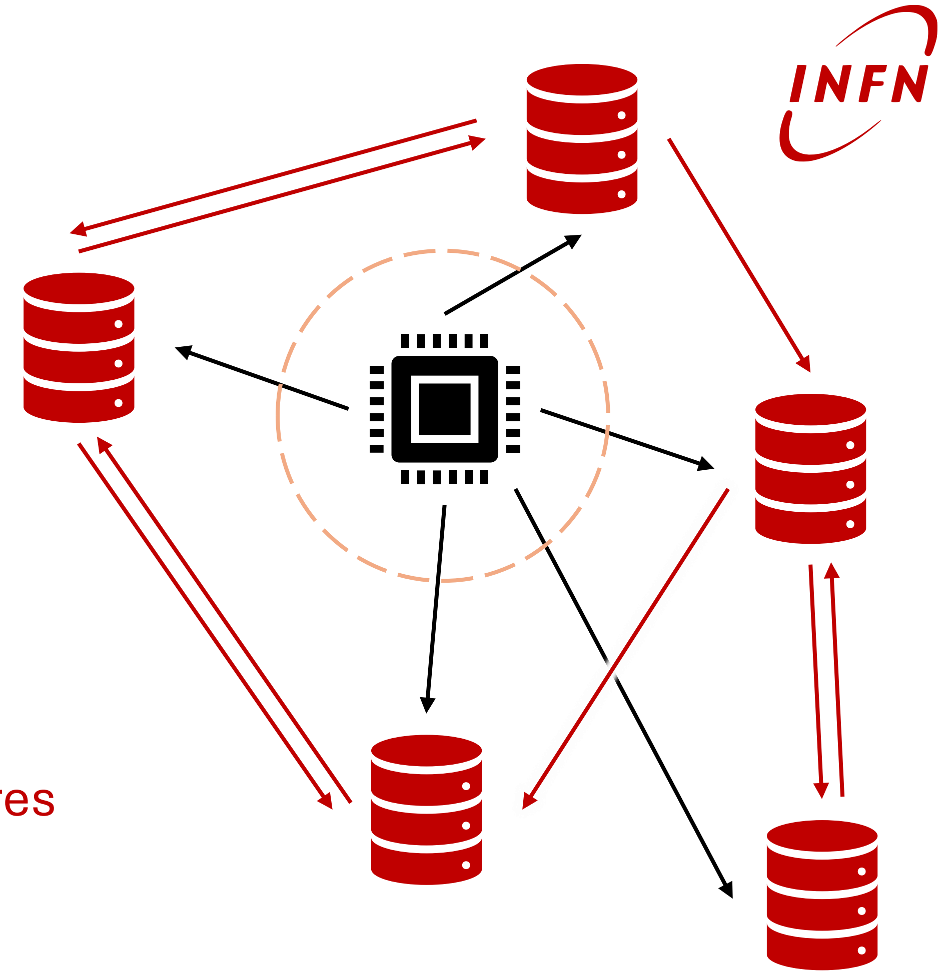
A joint effort (the case of INFN)

Physics:
- harness high-level quantum entropy
- ensuring stability & avoiding bias

Cryptography:
- primitives for extraction & expansion
- security claims

Mathematics:
- statistical validation
- min-entropy estimation

Systems engineering:
- secure distribution in large scale infrastructures
- high availability



Thanks!