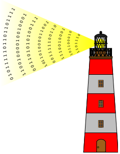
Randomness Beacons for Enhanced Public Auditability

and some notes on cryptography at NIST

Luís Brandão and René Peralta

Cryptographic Technology Group
National Institute of Standards and Technology
(Gaithersburg, Maryland, USA)



Presentation at Faculdade de Ciências, Universidade de Lisboa
Talks @ DI: Hosted by Departamento de Informática, FCUL
February 17, 2020 @ Lisboa, Portugal

* Part of a series of talks promoting the NIST project on Interoperable Randomness Beacons. Many slides are reused or adapted from previous presentations.

Outline

1. NIST and its crypto group
2. Randomness beacons — introduction
3. Randomness beacons — format, operations, use
4. Applications
5. Concluding remarks

Outline 1

Some NIST data

National Institute of Standards and Technology (NIST)

(National Bureau of Standards 1901–1988 → NIST 1988–present)

- **Mission** (keywords): innovation, industrial competitiveness, measurement science, standards and technology, economic security, quality of life.

Aerial photo of Gaithersburg campus (source: Google Maps, August 2019)

Some NIST data

National Institute of Standards and Technology (NIST)

(National Bureau of Standards 1901–1988 → NIST 1988–present)

- **Mission** (keywords): innovation, industrial competitiveness, measurement science, standards and technology, economic security, quality of life.

Wide spectrum of competences

- $\sim 6\text{–}7 \times 10^3$ workers
- Five laboratories and two centers
- Laboratories → Divisions → Groups → Projects
- Standards, research and applications

Aerial photo of Gaithersburg campus (source: Google Maps, August 2019)

Some NIST data

National Institute of Standards and Technology (NIST)

(National Bureau of Standards 1901–1988 → NIST 1988–present)

- **Mission** (keywords): innovation, industrial competitiveness, measurement science, standards and technology, economic security, quality of life.

Wide spectrum of competences

- $\sim 6\text{–}7 \times 10^3$ workers
- Five laboratories and two centers
- Laboratories → Divisions → Groups → Projects
- Standards, research and applications

Aerial photo of Gaithersburg campus (source: Google Maps, August 2019)

*Five laboratories:
- Communications Technology
- Engineering
- Information Technology
- Material Measurement
- Physical Measurement

*Two centers:
- Neutron Research;
- Nanoscale Science and Technology

Laboratories, divisions, groups

Information Technology Laboratory (ITL):

advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

Laboratories, divisions, groups

Information Technology Laboratory (ITL):

advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

Laboratories, divisions, groups

Information Technology Laboratory (ITL):

advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

→ **Cryptographic Technology Group (CTG):** research, develop, engineer, and produce guidelines, recommendations and best practices for cryptographic algorithms, methods, and protocols.

Laboratories, divisions, groups

Information Technology Laboratory (ITL):

advancing measurement science, standards, and technology through research and development in information technology, mathematics, and statistics.

→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

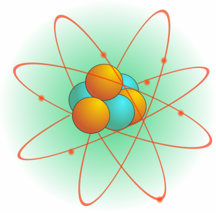
→ **Cryptographic Technology Group (CTG):** research, develop, engineer, and produce guidelines, recommendations and best practices for cryptographic algorithms, methods, and protocols.

- Documents: FIPS, SP 800, NISTIR.
- International cooperation: government, industry, academia, standardization bodies.

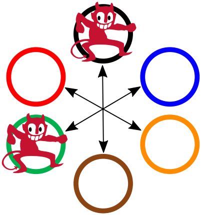
FIPS = Federal Information Processing Standards; SP 800 = Special Publications in Computer Security; NISTIR = NIST Internal or Interagency Report.

Some projects at the Crypto group (1/2)

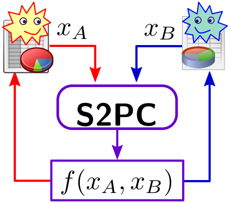
Post-Quantum Cryptography (PQC)



Threshold Cryptography (TC)



Privacy-Enhancing Cryptography (PEC)



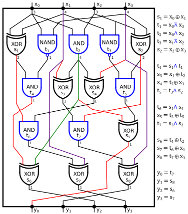
- **PQC:** new signatures; new PK-Encryption; 26 candidates after round 2; various math assumptions.
- **TC:** k-of-n threshold schemes; single-device and multi-party; f-of-n intrusion tolerance; resistance to side-channel attacks; validation.
- **PEC:** SMPC; ZKPs; develop reference material;

Some projects at the Crypto group (2/2)

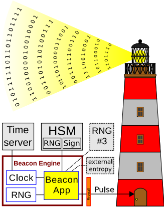
Lightweight Cryptography (LWC)

© nist.gov/image/lightweight2019png

Circuit Complexity (CC)



Interoperable-Randomness Beacons (IRBs)

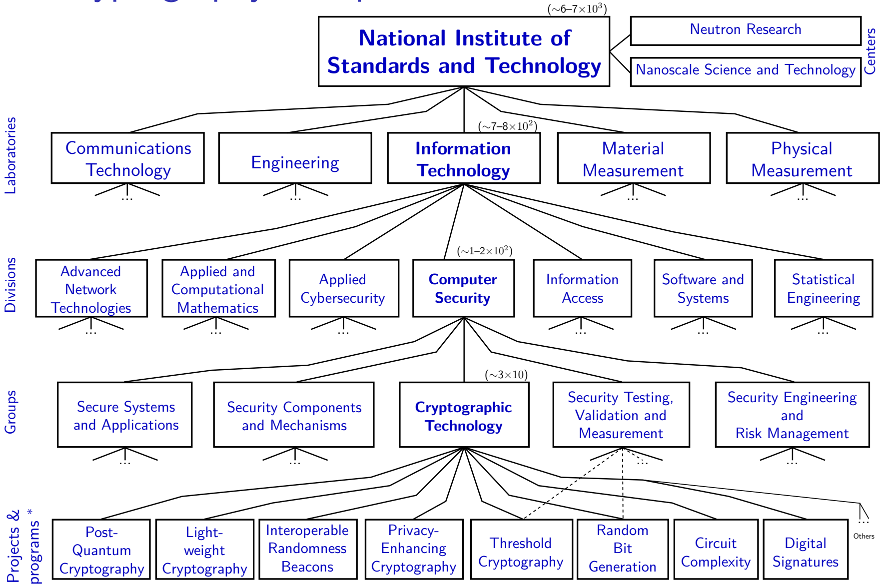


- **LWC:** symmetric crypto primitives; AEAD; 32 candidates after round 1
- **CC:** multiplicative complexity; symmetric functions; Karatsuba relations; ...
- **IRBs:** (this presentation)

Others, such as:

- **Signatures:** (FIPS 186-5) RSA, ECDSA, EdDSA
- **Random Bit Generation:** RNGs, PRNGs, testing and validation

The Cryptography Group at NIST



* (Some projects/programs involve several groups, divisions or laboratories)

(In parenthesis: approximate range # workers, inc. associates and fed. employees)

Collaborating with NIST

NIST is under the U.S. Department of Commerce.

The workforce includes many foreign guest researchers.

Let us know if you are interested in research collaborating / interning at CSD.

Outline 2

1. NIST and its crypto group
2. Randomness beacons — introduction
3. Randomness beacons — format, operations, use
4. Applications
5. Concluding remarks

Some concepts in this presentation

- Public randomness as a public good
- Randomness beacons for enhanced public auditability

Some concepts in this presentation

- Public randomness as a public good
- Randomness beacons for enhanced public auditability

At a high level

- **Randomness**
- **Public Good**
- **Audit**

Some concepts in this presentation

- Public randomness as a public good
- Randomness beacons for enhanced public auditability

At a high level (from Wikipedia):

- **Randomness:** "the lack of pattern or predictability in events [...] a measure of uncertainty of an outcome"
- **Public Good**
- **Audit**

Some concepts in this presentation

- Public randomness as a public good
- Randomness beacons for enhanced public auditability

At a high level (from Wikipedia):

- **Randomness:** "the lack of pattern or predictability in events [...] a measure of uncertainty of an outcome"

- **Public Good:** "a good [for which] individuals cannot be excluded from use, [and] use by one individual does not reduce availability to others."

- **Audit**

Some concepts in this presentation

- Public randomness as a public good
- Randomness beacons for enhanced public auditability

At a high level (from Wikipedia):

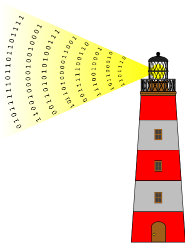
- **Randomness:** "the lack of pattern or predictability in events [...] a measure of uncertainty of an outcome"
- **Public Good:** "a good [for which] individuals cannot be excluded from use, [and] use by one individual does not reduce availability to others."
- **Audit:** "a systematic and independent examination [...] to ascertain how far the [...] statements [...] present a true and fair view [...]"

A Randomness Beacon

A Randomness Beacon

A service that produces timed outputs of fresh public randomness

(The idea goes back at least till 1983 — proposed by Rabin to aid crypto operations.)



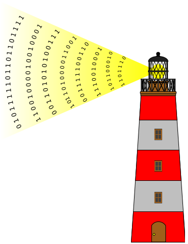
A Randomness Beacon

A service that produces timed outputs of fresh public randomness

(The idea goes back at least till 1983 — proposed by Rabin to aid crypto operations.)

At a high level:

- Periodically *pulsates* randomness
- Each pulse has a fresh 512-bit random string
- Each pulse is indexed, time-stamped and signed
- Any past pulse is publicly accessible
- The sequence of pulses forms a hash-chain



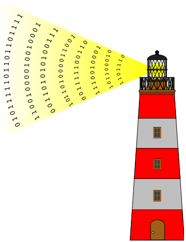
A Randomness Beacon

A service that produces timed outputs of fresh public randomness

(The idea goes back at least till 1983 — proposed by Rabin to aid crypto operations.)

At a high level:

- Periodically *pulsates* randomness
- Each pulse has a fresh 512-bit random string
- Each pulse is indexed, time-stamped and signed
- Any past pulse is publicly accessible
- The sequence of pulses forms a hash-chain



Can be useful for:

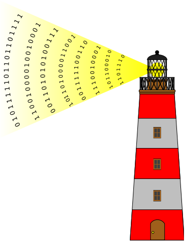
- public auditability of randomized processes
- coordination between multiple parties (e.g., who does/wins something)
- prove something happened after a certain time
- …

A Randomness Beacon

A service that produces timed outputs of fresh public randomness

(The idea goes back at least till 1983 — proposed by Rabin to aid crypto operations.)

At a high level:



- Periodically *pulsates* randomness
- Each pulse has a fresh 512-bit random string
- Each pulse is indexed, time-stamped and signed
- Any past pulse is publicly accessible
- The sequence of pulses forms a hash-chain

Can be useful for:

- public auditability of randomized processes
- coordination between multiple parties (e.g., who does/wins something)
- prove something happened after a certain time
- ...

NOT good for: selecting your secret keys

NIST project: Interoperable Randomness Beacons

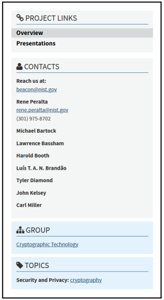
https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

NIST project: Interoperable Randomness Beacons

https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

The project has several tracks:

- **A.** promote a reference for randomness beacons;
- **B.** maintain a NIST Beacon implementation;
- **C.** promote the deployment of Beacons by multiple independent organizations;
- **D.** promote usages of beacon-issued randomness
- **E.** assist initiatives about trusted randomness, e.g., quantum RNGs and certifiable randomness.

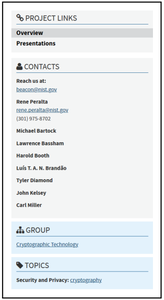


NIST project: Interoperable Randomness Beacons

https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

The project has several tracks:

- **A.** promote a reference for randomness beacons;
- **B.** maintain a NIST Beacon implementation;
- **C.** promote the deployment of Beacons by multiple independent organizations;
- **D.** promote usages of beacon-issued randomness
- **E.** assist initiatives about trusted randomness, e.g., quantum RNGs and certifiable randomness.



Some milestones:

- 2013: Prototype NIST beacon v1.0
- 2018: Quantum RNG by Physics Measurement Lab
- 2018: Deployment of NIST beacon v2.0
- 2019: Publication of Reference for randomness beacons

Example of a potential application

Example of a potential application

- Public officials are randomly selected for financial audits.
- The selected persons want to confirm how the selection was made.
- Citizens are also interested in verifying the random selection.

Example of a potential application

- Public officials are randomly selected for financial audits.
- The selected persons want to confirm how the selection was made.
- Citizens are also interested in verifying the random selection.
- The University of Chile is developing an application for selections based on public randomness from a Beacon.

Example of a potential application

- Public officials are randomly selected for financial audits.
- The selected persons want to confirm how the selection was made.
- Citizens are also interested in verifying the random selection.
- The University of Chile is developing an application for selections based on public randomness from a Beacon.

Security aspects

- Can the beacon be influenced to select (or not select) a particular official?
- Can an attacker learn in advance which officials will be selected?

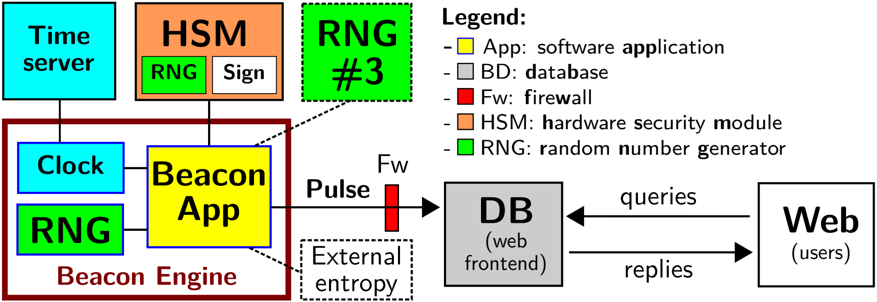
Example of a potential application

- Public officials are randomly selected for financial audits.
- The selected persons want to confirm how the selection was made.
- Citizens are also interested in verifying the random selection.
- The University of Chile is developing an application for selections based on public randomness from a Beacon.

Security aspects

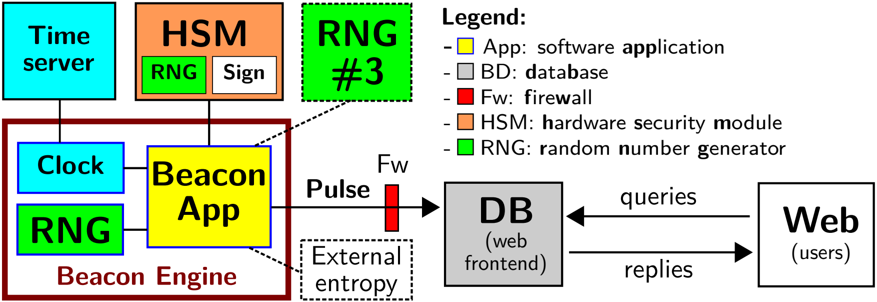
- Can the beacon be influenced to select (or not select) a particular official?
- Can an attacker learn in advance which officials will be selected?
- What interests are at stake? What resources does an adversary have?

Architecture of the NIST Beacon service



Legend:
- App: software **appl**ication
- BD: **d**ata**b**ase
- Fw: **f**ire**w**all
- HSM: **h**ardware **s**ecurity **m**odule
- RNG: **r**andom **n**umber **g**enerator

Architecture of the NIST Beacon service



A **Reference** for Randomness Beacons: Format and Protocol Version 2

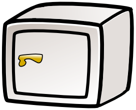
doi:10.6028/NIST.IR.8213-draft

Outline 3

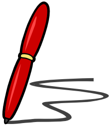
Some concepts useful in this talk

- **Hash:**

- **Commitment:**



- **[Digital] Signature:**



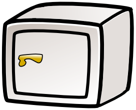
Some concepts useful in this talk

- **Hash:**

  — like a fingerprint of data (‘unique’ string 512 of bits)
  — looks random if its originator data is unknown

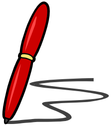
- **Commitment:**

  — like a vault that hides data, until it is opened
  — once closed, cannot change what is inside



- **[Digital] Signature:**

  — like a physical signature, but cannot be forged
  — a signature copied to another document is invalid



A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed
- Two main random values ("rands"): `randLocal` and `randOut`.

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed
- Two main random values (“rands”): randLocal and randOut.
- Other features: signed

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed
- Two main random values (“rands”): `randLocal` and `randOut`.
- Other features: signed, committed `randLocal`

A pulse (simplified example)

```
[1]  uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
[2]  version:str="2.0"
...  ...
[4]  period:dec="60000"
...  ...
[6]  chainId:dec="1"
[7]  pulseId:dec="220394"
[8]  time:str="2018-12-26T16:07:00.000Z"
[9]  randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
...  ...
[13] out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...  ...
[18] preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
...  ...
[20] sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
[21] randOut:hex="0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed
- Two main random values (“rands”): `randLocal` and `randOut`.
- Other features: signed, committed `randLocal`, chained `randOut`, ...

The two "rands" in a pulse

The two "rands" in a pulse

randLocal (local random value):

randOut (output value):

The two "rands" in a pulse

randLocal (local random value):

- Hash of randomness produced by ≥ 2 RNGs
- **Pre-committed** 1 minute in advance of release
- The PreCom of randLocal is the source of freshness for each pulse
- Useful for combining beacons

randOut (output value):

The two "rands" in a pulse

randLocal (local random value):

- Hash of randomness produced by ≥ 2 RNGs
- **Pre-committed** 1 minute in advance of release
- The PreCom of randLocal is the source of freshness for each pulse
- Useful for combining beacons

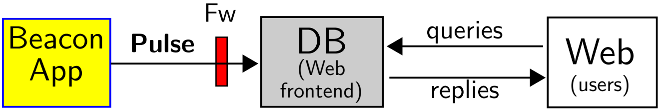
randOut (output value):

- Hash of all other fields
- **Fresh** at the time of release
- The randomness to be used by applications

Fetching pulses

Fetching pulses

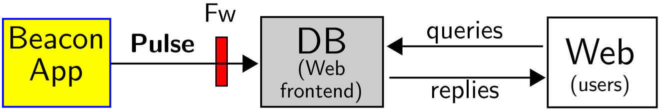
Beacon App: a pulse release means sending it to the database



Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

Fetching pulses

Beacon App: a pulse release means sending it to the database



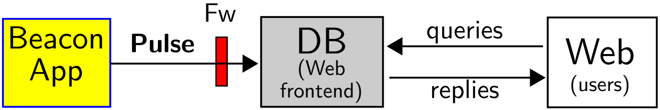
Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

The users request a pulse from the database through a URI/URL:

(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

Fetching pulses

Beacon App: a pulse release means sending it to the database



Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

The users request a pulse from the database through a URI/URL:

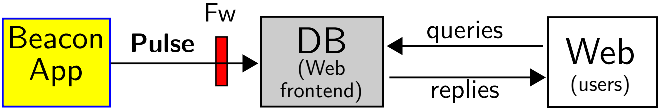
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

https://beacon.nist.gov/beacon/2.0/chain/last/pulse/last

Example: URL for the latest pulse in chain 1 of the NIST randomness Beacon (version 2)

Fetching pulses

Beacon App: a pulse release means sending it to the database



Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

The users request a pulse from the database through a URI/URL:

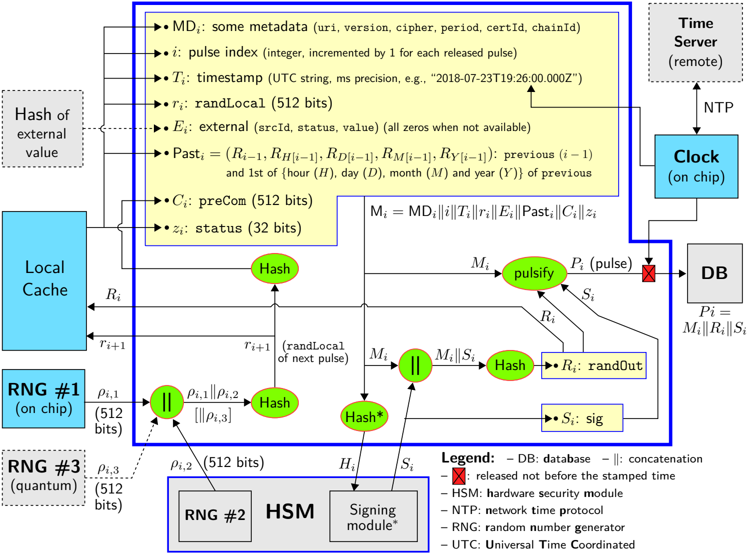
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

https://beacon.nist.gov/beacon/2.0/chain/last/pulse/last

Example: URL for the latest pulse in chain 1 of the NIST randomness Beacon (version 2)

Other queries exist: by pulseId; skiplists; certificates; external values...

A possible diagram of pulse generation



For simplicity, the diagram omits serialization details (e.g., field lengths and padding) and some metadata fields.

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$**:**

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

If I want to allow future auditability of a randomized operation:

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

If I want to allow future auditability of a randomized operation:

1. **Commit upfront:**
2. **Derive a seed:**
3. **Perform the operation:**

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

If I want to allow future auditability of a randomized operation:

1. **Commit upfront:** publish a statement S that explains my deterministic operation that will use the Beacon randomness (the output value `randOut`) from future time t;
2. **Derive a seed:**
3. **Perform the operation:**

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

If I want to allow future auditability of a randomized operation:

1. **Commit upfront:** publish a statement S that explains my deterministic operation that will use the Beacon randomness (the output value `randOut`) from future time t;
2. **Derive a seed:** Get $R =$ `randOut`$[t]$ (from the pulse with `timestamp` t), and set the seed as $Z = \text{Hash}(S||R)$
3. **Perform the operation:**

Using Beacon randomness (if I trust the Beacon)

(some simplifications for purpose of presentation)

Obtain a random integer within $[0, N-1]$:

- Just calculate `randOut` (mod N), if $N < 2^{384}$

If I want to allow future auditability of a randomized operation:

1. **Commit upfront:** publish a statement S that explains my deterministic operation that will use the Beacon randomness (the output value `randOut`) from future time t;
2. **Derive a seed:** Get $R =$ `randOut`$[t]$ (from the pulse with `timestamp` t), and set the seed as $Z = \text{Hash}(S||R)$
3. **Perform the operation:** Do what the statement S promised, using Z as the seed for all needed pseudo-randomness.

Do you need to trust the Beacon?

What happens if a malicious Beacon targets your application, to affect the unpredictability?

3 mitigations:

- Feed external entropy (external value field)
  – The Beacon cannot precompute randomness of the far away future
- Combine randomness from different beacons
  – No single beacon can affect the randomness that will be used
- Combine a local secret (and committed) value
  – The beacon cannot predict which seed the application will get

Some Beacons in development

Three countries are developing Beacons to match the current reference:

- (United States) NIST Randomness Beacon
  https://beacon.nist.gov/home
- (Chile) Random UChile
  https://beacon.clcert.cl/
- (Brazil) Brazilian Randomness Beacon
  https://beacon.inmetro.gov.br/

Some Beacons in development

Three countries are developing Beacons to match the current reference:

- (United States) NIST Randomness Beacon
  https://beacon.nist.gov/home
- (Chile) Random UChile
  https://beacon.clcert.cl/
- (Brazil) Brazilian Randomness Beacon
  https://beacon.inmetro.gov.br/

We would like others to join

Outline 4

Using beacon randomness

Choose a beacon (or multiple beacons).

Implement a beacon (an engineering task):

Use beacon randomness

Using beacon randomness

Choose a beacon (or multiple beacons). Certain societal application may require an official/certified beacon per jurisdiction, e.g., one per country.

Implement a beacon (an engineering task):

Use beacon randomness

Using beacon randomness

Choose a beacon (or multiple beacons). Certain societal application may require an official/certified beacon per jurisdiction, e.g., one per country.

Implement a beacon (an engineering task):

1. Follow the reference (NISTIR 8213)
2. Assemble the components and install the Beacon App
   (We plan to open-source release in 2020 the "NIST Beacon App" software.)
3. Ensure long-term availability; maintain equipment; prevent intrusions

Use beacon randomness

Using beacon randomness

Choose a beacon (or multiple beacons). Certain societal application may require an official/certified beacon per jurisdiction, e.g., one per country.

Implement a beacon (an engineering task):

1. Follow the reference (NISTIR 8213)
2. Assemble the components and install the Beacon App
   (We plan to open-source release in 2020 the "NIST Beacon App" software.)
3. Ensure long-term availability; maintain equipment; prevent intrusions

Use beacon randomness

- We can conceive many applications ... (next slides)
- It is up to the community to make them a reality.
- Some applications may require more fancy crypto (e.g., ZKPs)

Example applications:

- Select random test vs. control groups for clinical trials
- Select random government officials for financial audits
- Assign court cases to judges at random
- Sample random lots for quality-measuring procedures
- Provide entropy to digital lotteries
- Enable time-ordering evidence for audits in legal metrology

Some general objectives:

- Prevent auditors from biasing selections (or being accused of it)
- Prevent auditees from addressing only the to-be-sampled items
- Enable public verifiability of correct sampling

Example applications:

- Select random test vs. control groups for clinical trials
- Select random government officials for financial audits
- Assign court cases to judges at random
- Sample random lots for quality-measuring procedures
- Provide entropy to digital lotteries
- Enable time-ordering evidence for audits in legal metrology

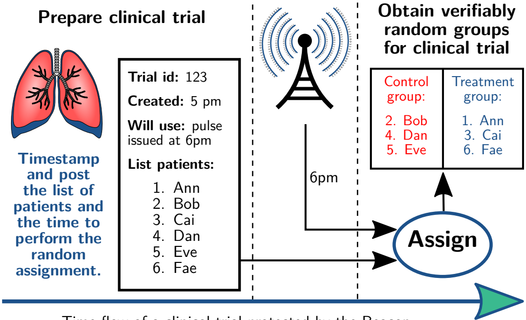
Some general objectives:

- Prevent auditors from biasing selections (or being accused of it)
- Prevent auditees from addressing only the to-be-sampled items
- Enable public verifiability of correct sampling

Advanced features: zero-knowledge proofs (ZKP) to enable auditability with privacy

Use case: randomized clinical trials

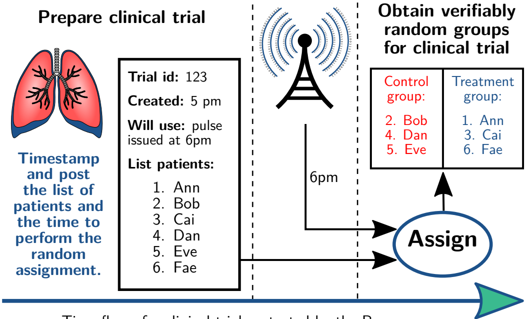
- **Setting:** a placebo-controlled clinical trial assigns patients to either the **treatment** group or the **control** group.
- **Goal:** After the study, it is possible to convince others that the trial was properly randomized.



Time flow of a clinical trial protected by the Beacon

Use case: randomized clinical trials

- **Setting:** a placebo-controlled clinical trial assigns patients to either the **treatment** group or the **control** group.
- **Goal:** After the study, it is possible to convince others that the trial was properly randomized.



Time flow of a clinical trial protected by the Beacon

Apply commitments and zero-knowledge proofs to hide private data while proving correctness.

Outline 5

1. NIST and its crypto group
2. Randomness beacons — introduction
3. Randomness beacons — format, operations, use
4. Applications
5. Concluding remarks

Concluding remarks

Concluding remarks

- Randomness Beacons have a **potential as public good/utility**, e.g., to enhance public auditability of randomized processes

Concluding remarks

- Randomness Beacons have a **potential as public good/utility**, e.g., to enhance public auditability of randomized processes

- The *reference* (NISTIR 8213) version 2 introduced new features for a better **interoperability, security and efficiency**

Concluding remarks

- Randomness Beacons have a **potential as public good/utility**, e.g., to enhance public auditability of randomized processes
- The *reference* (NISTIR 8213) version 2 introduced new features for a better **interoperability, security and efficiency**
- **We would like to have your collaboration:**
  - external apps using Beacon randomness
  - more deployed beacons

The test of time

70 years from now, will beacons (still) be used as a building block of public auditability?

The test of time

70 years from now, will beacons (still) be used as a building block of public auditability?

Photo in 1948 *

Photo in 2018: https://www.nist.gov/sites/default/files/documents/2018/06/15/nist_gaithersburg_master_plan_may_7_2018.pdf

The NIST Stone Test Wall: "Constructed [in 1948] to study the performance of stone subjected to weathering. It contains 2352 individual samples of stone, of which 2032 are domestic stone from 47 states, and 320 are stones from 16 foreign countries."

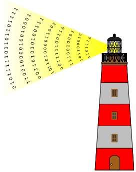
* https://www.nist.gov/el/materials-and-structural-systems-division-73100/nist-stone-wall

- NISTIR 8213: https://doi.org/10.6028/NIST.IR.8213-draft
- Beacon project: https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Thank you

- NISTIR 8213: https://doi.org/10.6028/NIST.IR.8213-draft
- Beacon project: https://csrc.nist.gov/Projects/Interoperable-Randomness-Beacons

Randomness Beacons for Enhanced Public Auditability



(and some notes on cryptography at NIST)

luis.brandao@nist.gov; rene.peralta@nist.gov

Presentation at Faculdade de Ciências, Universidade de Lisboa

Talks @ DI: Hosted by Departamento de Informática, FCUL

February 17, 2020 @ Lisboa, Portugal

List of slides

Some standardized cryptographic primitives

Traditional focus on "basic" primitives:

- Block ciphers: DES (1977), EES (1994), TDEA (1999), AES (2001)
- Cipher modes of operation (1980–): CBC, CT, CCM, GCM ...
- Hash functions (SHS): SHA-1 (1994), SHA-2 (2001), SHA-3 (2015)
- Signatures (DSS): DSA (1997), ECDSA (1998), RSA (2000), EdDSA (2019)
- Pair-wise key agreement, e.g., based on DH (2006) and RSA (2009)
- DRBGs (2006): CTR_, Hash_, HMAC_, Dual_EC_ (withdrawn in 2015 due to concerns of potential subversion)

(Not an exhaustive list; years indicated for perspective; some documentation has subsequent updates)
(Further details in "NIST Cryptographic Standards and Guidelines Development Program Briefing Book")

Some of these NIST-standards were specified with reference to standards by other bodies, and with further requirements.

Several methods:

- Internal or interagency developed techniques
- Adoption of external standards
- Open call, competition, "competition-like"

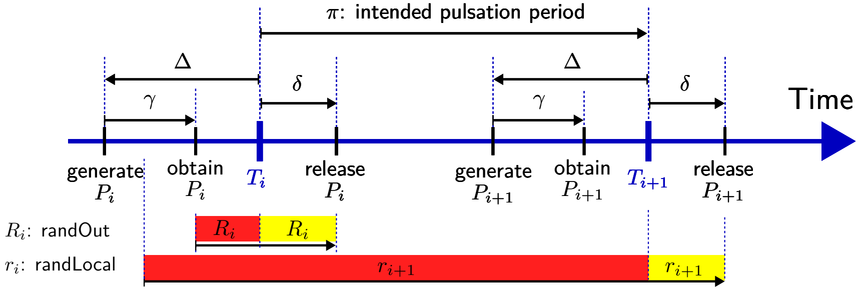
Legend:
- AES = Advanced Encryption Standard
- CBC = Cipher block chaining (mode)
- CT = Counter (mode)
- CCM = Counter with Cipher-block chaining
- DES = Data Encryption Standard
- DH = Diffie–Hellman
- DSA = Digital Signature Algorithm
- DSS = Digital Signature Standard
- DRBG = Deterministic Random Bit Generator
- ECDSA = Elliptic curve DSA
- EdDSA = Edwards curve DSA
- EES = Escrowed Encryption Standard
- GCM = Galois counter mode
- RSA = Rivest–Shamir–Adleman
- SHA = Secure Hash Algorithm
- SHS = Secure Hash Standard
- TDEA = Triple Data Encryption Algorithm

Timing for generation and release

1. No advanced release of pulse ($\delta \geq 0$)
2. Generate with entropy (≥ 2 RNGs)

} ⇒ **Unpredictability**

3. No advanced generation (small Δ) ⇒ **Freshness**
4. No delayed release (small δ) ⇒ **Timeliness**
5. Unambiguous indexation ⇒ **Unambiguity**



(The actual requirements specify allowed intervals for δ and Δ)

Use case: public auditability with privacy

Challenge: random selection depending on private attributes

Public		Private initial			Private derivative	
# (i)	Rand id	Name (N)	a	b	Weight (w)	Acc. (W)
1	371	**C**ai	1	2	0.1	0.1
2	942	**E**ve	2	7	0.3	0.4
3	107	**B**ob	1	5	0.2	0.6
4	527	**A**nn	1	9	0.3	0.9
5	123	**D**an	3	1	0.1	1.0

Commit to all attributes and publish the table of commitments ... then **prove in ZK**:

1. $a_i \in A$ (e.g., annual salary); $b_i \in B$ (e.g., years in position);
2. $w_i = f(a_i, b_i)$ (correct probability weight);
3. $\sum_i w_i = 1$ (correct sum of weights);
4. $W_i = w_i + W_{i-1}$ (correct accumulator);
5. $\{N_i\}$ =NAMES (non-repeated names from an appropriate set); ...

Derive $R : 0 < R \leq 1$ (random) from the Beacon and determine # j: $W_{\max(1,j-1)} < R \leq W_j$

- **Prove in ZK** that j is consistent with R and the table of commitments

Some references

See NISTIR 8213 for more references.

Examples of other approaches to randomness:

- 1981: M. Blum. *Coin flipping by telephone*.
- 1983: M. Rabin. *Transaction protection by beacons*.
- Combining public randomness from multiple sources:
  - The League of Entropy (Decentralized Randomness Beacon)
  - Other systems, e.g., RandHerd, RandHound, Scrape, HydRand
- Certifiable randomness, based on:
  - Bell tests (internally verifiable)
  - Quantum-supremacy demonstrations (externally verifiable)
- Random.org "True Random Number Service"