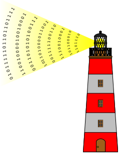
Randomness Beacons as Enablers of Public Auditability

Luís Brandão

Cryptographic Technology Group
National Institute of Standards and Technology



Presentation at STPPA 01
Special Topics on Privacy and Public Auditability
January 27, 2020 @ Gaithersburg, Maryland, USA

Some slides are based on previous presentations (IMFD Oct 2019, ICMC May 2019).

The Reference for Randomness Beacons is joint work with John Kelsey, Rene Peralta and Harlod Booth.

The Interoperable Randomness Beacons project is joint work with others in the Cryptographic Technology Group.

Outline

Outline

1. Introduction
2. Randomness Beacons (format and operations)
3. Usages of beacon randomness
4. Concluding remarks

Goals of this presentation:

- Brief overview of the NIST Reference for Randomness Beacons
- Allude to possible public-auditability applications

Outline 1

1. Introduction
2. Randomness Beacons (format and operations)
3. Usages of beacon randomness
4. Concluding remarks

Some concepts in this presentation

Some concepts in this presentation

At a high level (from Wikipedia):

Randomness

Public Good

Audit

Some concepts in this presentation

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

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

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

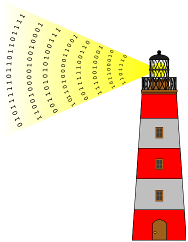
- "a systematic and independent examination [...] to ascertain how far the [...] statements [...] present a true and fair view [...]"

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.)



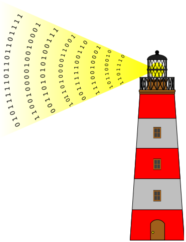
A Randomness Beacon

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



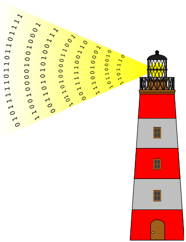
A Randomness Beacon

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



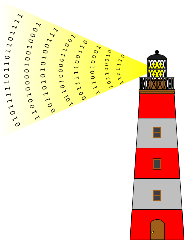
A Randomness Beacon

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



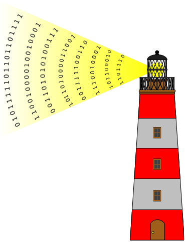
A Randomness Beacon

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



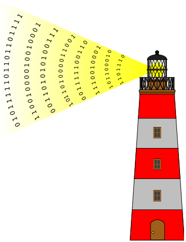
A Randomness Beacon

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



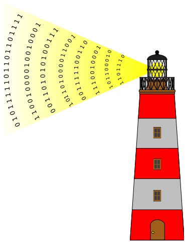
A Randomness Beacon

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

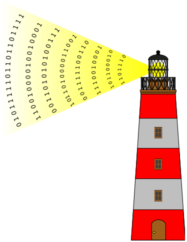
- public auditability of randomized processes
- coordination between multiple parties (e.g., who does/wins something)
- prove something happened after a certain time
- ...

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

An example/conceivable application

An example/conceivable application

- A tax Comptroller selects, at random, public officials for financial audit.
- The selected person want to confirm how the selection was made.
- A citizen at home also wants to see a proof of random selection.

An example/conceivable application

- A tax Comptroller selects, at random, public officials for financial audit.
- The selected person want to confirm how the selection was made.
- A citizen at home also wants to see a proof of random selection.
- The University of Chile is developing an application for selections based on public randomness from a Beacon.

An example/conceivable application

- A tax Comptroller selects, at random, public officials for financial audit.
- The selected person want to confirm how the selection was made.
- A citizen at home also wants to see a proof of 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?

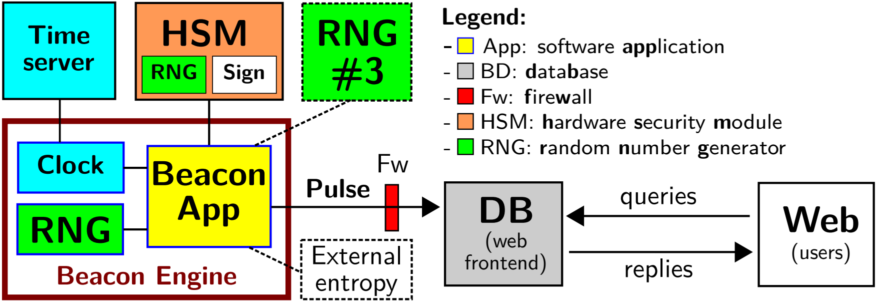
An example/conceivable application

- A tax Comptroller selects, at random, public officials for financial audit.
- The selected person want to confirm how the selection was made.
- A citizen at home also wants to see a proof of random selection.
- The University of Chile is developing an application for selections based on public randomness from a Beacon.

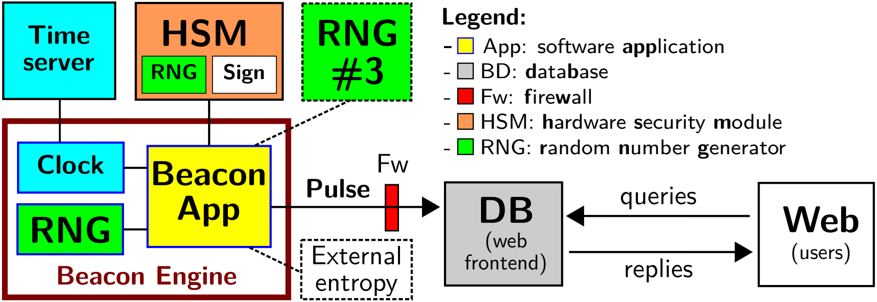
Security aspects

- 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 Beacon service

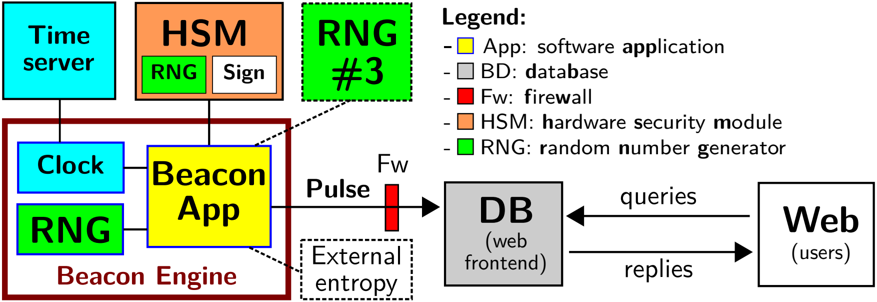


Architecture of the Beacon service



But, what exactly is a *pulse*? where does its randomness come from?, ...

Architecture of the Beacon service

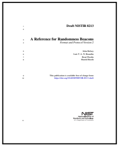


But, what exactly is a *pulse*? where does its randomness come from?, ...

A **Reference** for Randomness Beacons:

Format and Protocol Version 2

https://doi.org/10.6028/NIST.IR.8213-draft



NIST project: Interoperable Randomness Beacons

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

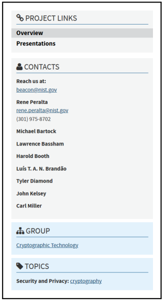
NIST project: Interoperable Randomness Beacons

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

The project has four main 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

Also interested in assisting initiatives about trusted randomness, e.g., quantum RNGs and certifiable randomness.



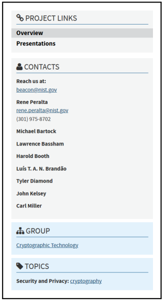
NIST project: Interoperable Randomness Beacons

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

The project has four main 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

Also interested in assisting initiatives about trusted randomness, e.g., quantum RNGs and certifiable randomness.



Some milestones:

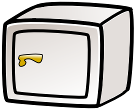
- 2013: Prototype NIST beacon v1.0
- 2018: Quantum RNG by Physics Measurement Lab
- 2018: Deployment of NIST beacon v2.0
- 2019: Publication of Reference for randomness beacons

Outline 2

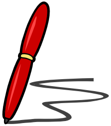
Some concepts useful in this talk

- **Hash:**

- **Commitment:**



- **[Digital] Signature:**



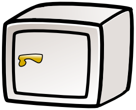
Some concepts useful in this talk

- **Hash:**

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

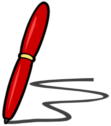
- **Commitment:**

  — 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):

- **What:** Hash of randomness produced by ≥ 2 RNGs
- **How:** **Pre-committed** 1 minute in advance of release
- **Why:** Randomness contribution to combine with randomness of other beacons

randOut (output value):

The two “rands” in a pulse

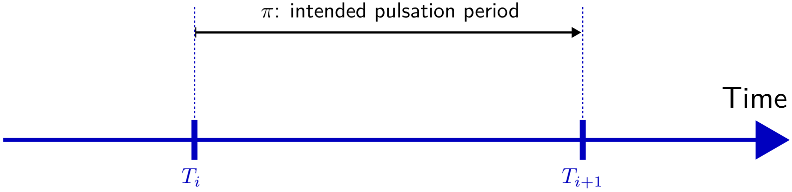
randLocal (local random value):

- **What:** Hash of randomness produced by ≥ 2 RNGs
- **How:** **Pre-committed** 1 minute in advance of release
- **Why:** Randomness contribution to combine with randomness of other beacons

randOut (output value):

- **What:** Hash of all other fields
- **How:** **Fresh** at the time of release
- **Why:** Randomness seed for applications that completely trust this beacon

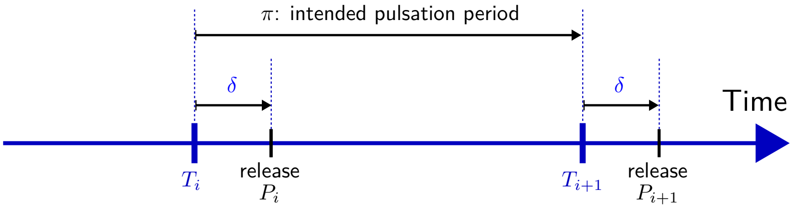
Timing for generation and release



Timing for generation and release

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

} $\Rightarrow$ **Unpredictability**

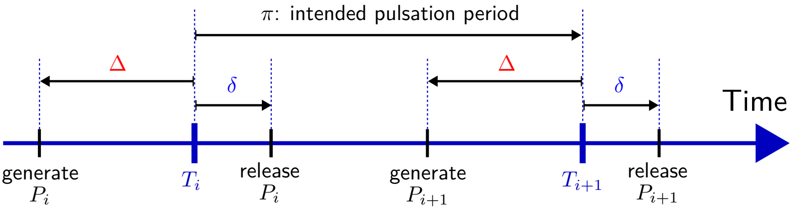


Timing for generation and release

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

} ⇒ **Unpredictability**

3. No advanced generation (small Δ) ⇒ **Freshness**

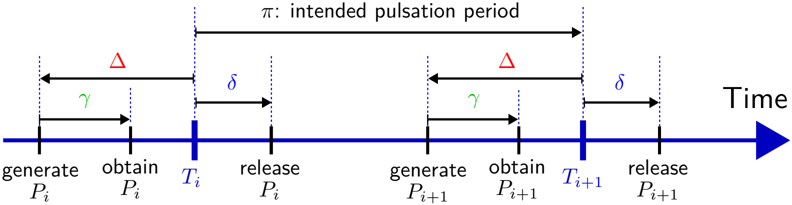


Timing for generation and release

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

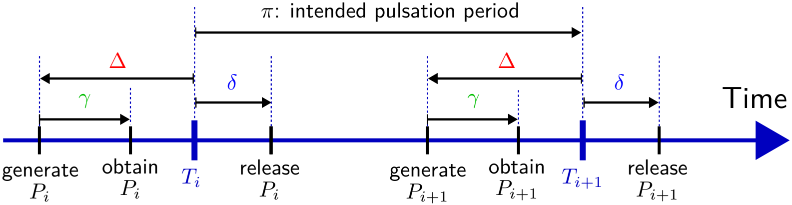
} $\Rightarrow$ **Unpredictability**

3. No advanced generation (small Δ) $\Rightarrow$ **Freshness**
4. No delayed release (small γ and δ) $\Rightarrow$ **Timeliness**



Timing for generation and release

1. No advanced release of pulse ($\delta \geq 0$) } ⇒ **Unpredictability**
2. Generate with entropy (≥ 2 RNGs) } ⇒ **Unpredictability**
3. No advanced generation (small Δ) ⇒ **Freshness**
4. No delayed release (small γ and δ) ⇒ **Timeliness**
5. Unambiguous indexation ⇒ **Unambiguity**

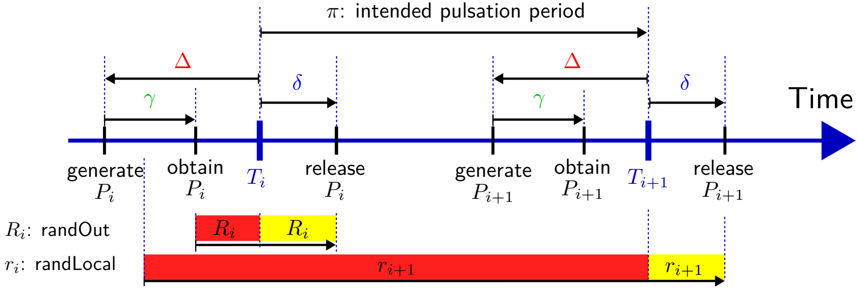


Timing for generation and release

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

} $\Rightarrow$ **Unpredictability**

3. No advanced generation (small Δ) $\Rightarrow$ **Freshness**
4. No delayed release (small γ and δ) $\Rightarrow$ **Timeliness**
5. Unambiguous indexation $\Rightarrow$ **Unambiguity**

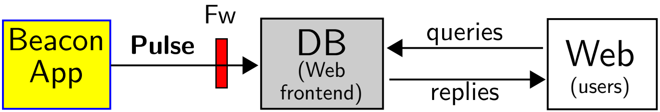


(The actual requirements specify allowed intervals for δ and Δ)

Fetching pulses

Fetching pulses

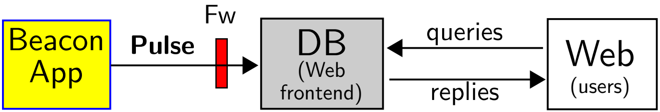
Beacon App: a pulse release means sending it to the database



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



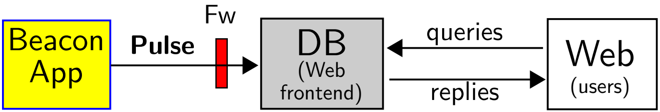
Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

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:

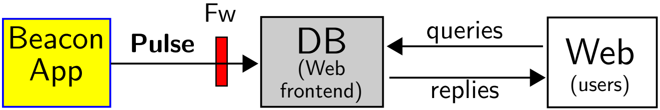
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

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:

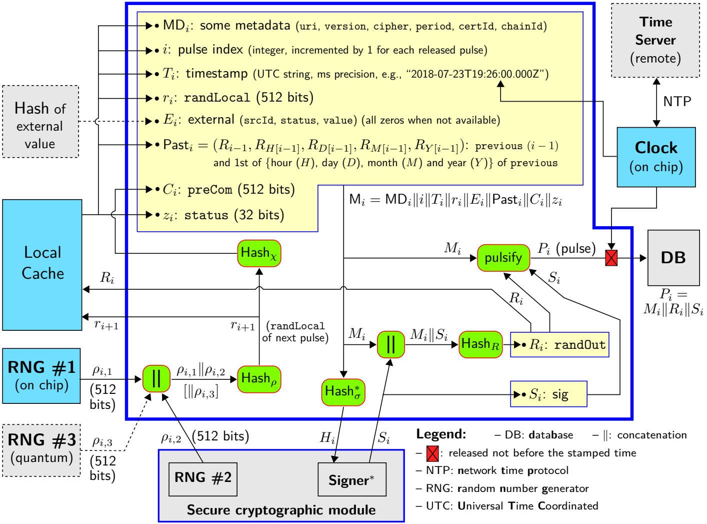
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

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.

Outline 3

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 (e.g., the Comptroller), to affect the unpredictability?

Do you need to trust the Beacon?

What happens if a malicious Beacon targets your application (e.g., the Comptroller), to affect the unpredictability?

3 mitigations:

Do you need to trust the Beacon?

What happens if a malicious Beacon targets your application (e.g., the Comptroller), to affect the unpredictability?

3 mitigations:

- Feed external entropy (external value field)
  – The Beacon cannot precompute randomness of the far away future

Do you need to trust the Beacon?

What happens if a malicious Beacon targets your application (e.g., the Comptroller), 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 various beacons
  – No single beacon can affect the randomness that will be used

Do you need to trust the Beacon?

What happens if a malicious Beacon targets your application (e.g., the Comptroller), 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 various 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

Some conceivable applications

"*You have been randomly selected for additional screening*"

Some conceivable applications

"*You have been randomly selected for additional screening*"

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 conceivable applications

"*You have been randomly selected for additional screening*"

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

Some conceivable applications

"*You have been randomly selected for additional screening*"

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: 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

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

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); ...

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**:

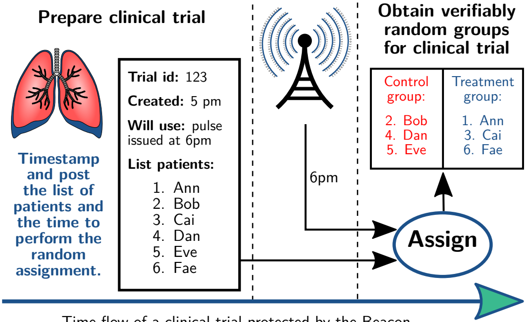
1. $a_i \in A$ (e.g., annual salary); $b_i \in B$ (e.g., years in position);
2. $w_i = f(a_i, b_i)$ (correct probability weight);
3. $\sum_i w_i = 1$ (correct sum of weights);
4. $W_i = w_i + W_{i-1}$ (correct accumulator);
5. $\{N_i\}$ =NAMES (non-repeated names from an appropriate set); ...

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

Use case: randomized clinical trials

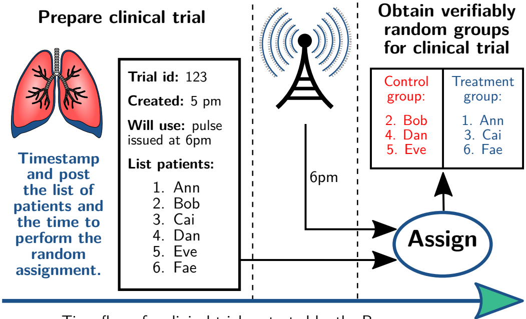
- **Setting:** a placebo-controlled clinical trial assigns patients to either the **treatment** group or the **control** group.
- **Goal:** After the study, it is possible to convince others that the trial was properly randomized.



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 4

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 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 better **interoperability, security and efficiency**
- Numerous stakeholders; applications can be reused across beacons.

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 better **interoperability, security and efficiency**
- Numerous stakeholders; applications can be reused across beacons.
- Some challenges and to-dos:
  - Learn exact requirements and constraints for concrete applications
  - Develop complementary analysis and guidance
  - Support deployment in multiple organizations
  - (Technical advances: post-quantum; period vs. pre-commitment; ...)

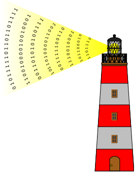
Concluding remarks

- 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 better **interoperability, security and efficiency**
- Numerous stakeholders; applications can be reused across beacons.
- Some challenges and to-dos:
  - Learn exact requirements and constraints for concrete applications
  - Develop complementary analysis and guidance
  - Support deployment in multiple organizations
  - (Technical advances: post-quantum; period vs. pre-commitment; ...)
- **We would like to have your collaboration!**

- 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 as Enablers of Public Auditability

Presentation at Special Topics on Privacy and Public Auditability

January 27, 2020 @ Gaithersburg, Maryland, USA

luis.brandao@nist.gov

List of slides