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

1. Introduction

2. Randomness Beacons (format and operations)

3. Usages of beacon randomness

4. Concluding remarks
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

1. Introduction

Some concepts in this presentation

1. Introduction

Some concepts in this presentation

1. Introduction
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 wants 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

Legend:
- ▶️ App: software application
- □ BD: database
- □ Fw: firewall
- □ HSM: hardware security module
- ▶️ RNG: random number generator

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

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

This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8213-draft
Architecture of the Beacon service

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

Architecture of the Beacon service

Legend:
- App: software application
- BD: database
- Fw: firewall
- HSM: hardware security module
- RNG: random number generator

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


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.
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

1. Introduction

2. Randomness Beacons (format and operations)

3. Usages of beacon randomness

4. Concluding remarks
Some concepts useful in this talk

- **Hash:**
  - like a fingerprint of data ('unique' string of 512 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
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)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>uri:str=&quot;<a href="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394">https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394</a>&quot;</td>
</tr>
<tr>
<td>2</td>
<td>version:str=&quot;2.0&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>period:dec=&quot;60000&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>chainId:dec=&quot;1&quot;</td>
</tr>
<tr>
<td>7</td>
<td>pulseId:dec=&quot;220394&quot;</td>
</tr>
<tr>
<td>8</td>
<td>time:str=&quot;2018-12-26T16:07:00.000Z&quot;</td>
</tr>
<tr>
<td>9</td>
<td>randLocal:hex=&quot;5FF1E0C019C42C77FA72D522... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>out.Prev:hex=&quot;BA646CC4E7AE195D2C85E9D3... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>preCom:hex=&quot;269908B840E79BE71CEC4EBA... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>sig:hex=&quot;17943D886DA8C7C24B9244BE... (4096 bits total)&quot;</td>
</tr>
<tr>
<td>21</td>
<td>randOut:hex=&quot;0A8863E03E200F6940A009B0... (512 bits total)&quot;</td>
</tr>
</tbody>
</table>
A pulse (simplified example)

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

- Each pulse is indexed
A pulse (simplified example)

```
[2] version: str="2.0"
... 
[4] period: dec="60000"
... 
[6] chainId: dec="1"
pulseId: dec="220394"
[8] time: str="2018-12-26T16:07:00.000Z"
[9] randLocal: hex="5FF1E0C019C42C77FA72D522...(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)

```
uri: str = "https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
version: str = "2.0"
...

period: dec = "60000"
...

chainId: dec = "1"
pulseId: dec = "220394"

time: str = "2018-12-26T16:07:00.000Z"
randLocal: hex = "5FF1E0C019C42C77FA72D522...(512 bits total)"
...
out.Prev: hex = "BA646CC4E7AE195D2C85E9D3...(512 bits total)"
...
preCom: hex = "269908B840E79BE71CEC4EBA...(512 bits total)"
...
sig: hex = "17943D886DA8C7C24B9244BE...(4096 bits total)"
randOut: hex = "0A8863E03E200F6940A009B0...(512 bits total)"
```

- Each pulse is indexed
A pulse (simplified example)

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>uri: str=&quot;<a href="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394">https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394</a>&quot;</td>
</tr>
<tr>
<td>2</td>
<td>version: str=&quot;2.0&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>period: dec=&quot;60000&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>chainId: dec=&quot;1&quot;</td>
</tr>
<tr>
<td>7</td>
<td>pulseId: dec=&quot;220394&quot;</td>
</tr>
<tr>
<td>8</td>
<td>time: str=&quot;2018-12-26T16:07:00.000Z&quot;</td>
</tr>
<tr>
<td>9</td>
<td>randLocal: hex=&quot;5FF1E0C019C42C77FA72D522... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>out.Prev: hex=&quot;BA646CC4E7AE195D2C85E9D3... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>preCom: hex=&quot;269908B840E79BE71CEC4EBA... (512 bits total)&quot;</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>sig: hex=&quot;17943D886DA8C7C24B9244BE... (4096 bits total)&quot;</td>
</tr>
<tr>
<td>21</td>
<td>randOut: hex=&quot;0A8863E03E200F6940A009B0... (512 bits total)&quot;</td>
</tr>
</tbody>
</table>

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

```
uri:str="https://beacon.nist.gov/beacon/2.0/chain/1/pulse/220394"
version:str="2.0"
... 
period:dec="60000"
... 
chainId:dec="1"
pulseId:dec="220394"
time:str="2018-12-26T16:07:00.000Z"
randLocal:hex="5FF1E0C019C42C77FA72D522...(512 bits total)"
... 
out.Prev:hex="BA646CC4E7AE195D2C85E9D3...(512 bits total)"
... 
preCom:hex="269908B840E79BE71CEC4EBA...(512 bits total)"
... 
sig:hex="17943D886DA8C7C24B9244BE...(4096 bits total)"
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)

- Each pulse is indexed
- Two main random values ("rands"): randLocal and randOut.
- Other features: signed, committed randLocal
A pulse (simplified example)

```plaintext
[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)"
...  
...  
[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

\textbf{randLocal} (local random value):

\textbf{randOut} (output value):
The two “rands” in a pulse

**randLocal** (local random value):

- **What:** Hash of randomness produced by $\geq 2$ 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 $\geq 2$ 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

\[ T_i \cdot \pi \cdot T_{i+1} \]

\[ \delta, \Delta, \gamma \]

\[ \text{Unpredictability} \]

\[ \text{Freshness} \]

\[ \text{Timeliness} \]

\[ \text{Unambiguity} \]

\( R_i \cdot \text{randOut} \)

\( r_{i+1} \cdot \text{randLocal} \)
Timing for generation and release

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

\[ \Rightarrow \text{Unpredictability} \]
Timing for generation and release

1. No advanced release of pulse ($\delta \geq 0$)
2. Generate with entropy ($\geq 2$ RNGs)
3. No advanced generation (small $\Delta$) $\Rightarrow$ Freshness

\[
\begin{align*}
\text{generate } P_i & \quad T_i & \quad \text{release } P_i & \quad T_{i+1} & \quad \text{release } P_{i+1} \\
\Delta & \quad \delta & \quad \Delta & \quad \delta
\end{align*}
\]

$\pi$: intended pulsation period

$\Rightarrow$ Unpredictability

The actual requirements specify allowed intervals for $\delta$ and $\Delta$.
Timing for generation and release

1. No advanced release of pulse ($\delta \geq 0$) \[ \Rightarrow \text{Unpredictability} \]
2. Generate with entropy ($\geq 2$ RNGs)
3. No advanced generation (small $\Delta$) \[ \Rightarrow \text{Freshness} \]
4. No delayed release (small $\gamma$ and $\delta$) \[ \Rightarrow \text{Timeliness} \]
Randomness Beacons (format and operations)

Timing for generation and release

1. No advanced release of pulse \( (\delta \geq 0) \)  
2. Generate with entropy \( (\geq 2 \text{ RNGs}) \)  
3. No advanced generation (small \( \Delta \))  \( \Rightarrow \) Freshness  
4. No delayed release (small \( \gamma \) and \( \delta \))  \( \Rightarrow \) Timeliness  
5. Unambiguous indexation  \( \Rightarrow \) Unambiguity
Timing for generation and release

1. No advanced release of pulse \((\delta \geq 0)\) \} \Rightarrow \text{Unpredictability}
2. Generate with entropy \((\geq 2 \text{ RNGs})\)
3. No advanced generation (small \(\Delta\) \Rightarrow \text{Freshness}
4. No delayed release (small \(\gamma\) and \(\delta\)) \Rightarrow \text{Timeliness}
5. Unambiguous indexation \Rightarrow \text{Unambiguity}

\(\pi\): intended pulsation period
\(\Delta\): release interval
\(\gamma\): interval between generation and obtain
\(\delta\): interval between obtain and release

\(R_i\): randOut
\(r_i\): randLocal

(The actual requirements specify allowed intervals for \(\delta\) and \(\Delta\))
Fetching pulses
Fetching pulses

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

Legend: App: application; DB: database; Fw: firewall.
Fetching pulses

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

Legend: App: application; DB: database; Fw: firewall.

The users request a pulse from the database through a URI/URL:
(URI = uniform resource identifier; URL = uniform resource locator)
Fetching pulses

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

Legend: App: application; DB: database; Fw: firewall.

The users request a pulse from the database through a URI/URL:
(URI = uniform resource identifier; URL = uniform resource locator)

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: application; DB: database; Fw: firewall.

The users request a pulse from the database through a URI/URL:
(URI = uniform resource identifier; URL = uniform resource locator)

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 pulseld; skiplists; certificates; external values...
A possible diagram of pulse generation

- MD_i: some metadata (uri, version, cipher, period, certId, chainId)
- i: pulse index (integer, incremented by 1 for each released pulse)
- T_i: timestamp (UTC string, ms precision, e.g., “2018-07-23T19:26:00.000Z”)
- r_i: randLocal (512 bits)
- E_i: external (srcId, status, value) (all zeros when not available)
- Past_i = (R_{i-1}, R_H[i-1], R_D[i-1], R_M[i-1], R_Y[i-1]): previous (i-1) and 1st of {hour (H), day (D), month (M) and year (Y)} of previous
- C_i: preCom (512 bits)
- z_i: status (32 bits)
- M_i = MD_i||i||T_i||r_i||E_i||Past_i||C_i||z_i

For simplicity, the diagram omits serialization details (e.g., field lengths and padding) and some metadata fields.
Outline 3

1. Introduction

2. Randomness Beacons (format and operations)

3. Usages of beacon randomness

4. Concluding remarks
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 $\text{randOut} \pmod{N}$, 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 $\text{randOut} \pmod N$, 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 $\text{randOut} \pmod{N}$, 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 \(\text{randOut} \mod N\), 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 \(\text{randOut}\)) 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 $\text{randOut} \pmod{N}$, 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 $\text{randOut}$) from future time $t$;

2. **Derive a seed**: Get $R = \text{randOut}[t]$ (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 $\text{randOut} \pmod{N}$, 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 $\text{randOut}$) from future time $t$;

2. **Derive a seed**: Get $R = \text{randOut}[t]$ (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

<table>
<thead>
<tr>
<th># (i)</th>
<th>Rand id</th>
<th>Name (N)</th>
<th>a</th>
<th>b</th>
<th>Weight (w)</th>
<th>Acc. (W)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>371</td>
<td>Cai</td>
<td>1</td>
<td>2</td>
<td>0.1</td>
<td>0.1</td>
</tr>
<tr>
<td>2</td>
<td>942</td>
<td>Eve</td>
<td>2</td>
<td>7</td>
<td>0.3</td>
<td>0.4</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
<td>Bob</td>
<td>1</td>
<td>5</td>
<td>0.2</td>
<td>0.6</td>
</tr>
<tr>
<td>4</td>
<td>527</td>
<td>Ann</td>
<td>1</td>
<td>9</td>
<td>0.3</td>
<td>0.9</td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>Dan</td>
<td>3</td>
<td>1</td>
<td>0.1</td>
<td>1.0</td>
</tr>
</tbody>
</table>

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

1. $a_i \in A$ (e.g., annual salary);
2. $w_i = f(a_i, b_i)$ (correct probability weight);
3. $\sum w_i = 1$ (correct sum of weights);
4. $W_i = w_i + W_{i-1}$ (correct accumulator);
5. $\{N_i\} = \text{NAMES}$ (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: public auditability with privacy

Challenge: random selection depending on private attributes

<table>
<thead>
<tr>
<th>Public</th>
<th>Private initial</th>
<th>Private derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td># (i)</td>
<td>Rand id</td>
<td>Name (N)</td>
</tr>
<tr>
<td>1</td>
<td>371</td>
<td>Cai</td>
</tr>
<tr>
<td>2</td>
<td>942</td>
<td>Eve</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
<td>Bob</td>
</tr>
<tr>
<td>4</td>
<td>527</td>
<td>Ann</td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>Dan</td>
</tr>
</tbody>
</table>

Commit to all attributes and publish the table of commitments
Use case: public auditability with privacy

Challenge: random selection depending on private attributes

<table>
<thead>
<tr>
<th>Public</th>
<th>Private initial</th>
<th>Private derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td># (i)</td>
<td>Rand id</td>
<td>Name (N)</td>
</tr>
<tr>
<td>1</td>
<td>371</td>
<td>Cai</td>
</tr>
<tr>
<td>2</td>
<td>942</td>
<td>Eve</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
<td>Bob</td>
</tr>
<tr>
<td>4</td>
<td>527</td>
<td>Ann</td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>Dan</td>
</tr>
</tbody>
</table>

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. $\{N_i\} = \text{NAMES}$ (non-repeated names from an appropriate set); ...
Use case: public auditability with privacy

Challenge: random selection depending on private attributes

<table>
<thead>
<tr>
<th>Public</th>
<th>Private initial</th>
<th>Private derivative</th>
</tr>
</thead>
<tbody>
<tr>
<td># (i)</td>
<td>Rand id</td>
<td>Name (N)</td>
</tr>
<tr>
<td>1</td>
<td>371</td>
<td>Cai</td>
</tr>
<tr>
<td>2</td>
<td>942</td>
<td>Eve</td>
</tr>
<tr>
<td>3</td>
<td>107</td>
<td>Bob</td>
</tr>
<tr>
<td>4</td>
<td>527</td>
<td>Ann</td>
</tr>
<tr>
<td>5</td>
<td>123</td>
<td>Dan</td>
</tr>
</tbody>
</table>

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. \( \sum w_i = 1 \) (correct sum of weights);
4. \( W_i = w_i + W_{i-1} \) (correct accumulator);
5. \( \{N_i\} = \text{NAMES} \) (non-repeated names from an appropriate set); ...

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

▶ 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.

### Diagram:

- **Prepare clinical trial**
  - **Trial id:** 123
  - **Created:** 5 pm
  - **Will use:** pulse issued at 6 pm
  - **List patients:**
    1. Ann
    2. Bob
    3. Cai
    4. Dan
    5. Eve
    6. Fae

- **Obtain verifiably random groups for clinical trial**
  - **Control group:** 2. Bob 4. Dan 5. Eve
  - **Treatment group:** 1. Ann 3. Cai 6. Fae

---

**Timestamp and post the list of patients and the time to perform the random assignment.**

**Time flow of a clinical trial protected by the Beacon**

---

22/26
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.

```
Prepare clinical trial

Timestamp and post the list of patients and the time to perform the random assignment.

Trial id: 123
Created: 5 pm
Will use: pulse issued at 6pm
List patients:
1. Ann
2. Bob
3. Cai
4. Dan
5. Eve
6. Fae

Obtain verifiably random groups for clinical trial

Control group:
2. Bob
4. Dan
5. Eve

Treatment group:
1. Ann
3. Cai
6. Fae

Assign

6pm

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

1. Introduction

2. Randomness Beacons (format and operations)

3. Usages of beacon randomness

4. 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 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; ...)

We would like to have your collaboration!
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!
4. Concluding remarks

- 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

Disclaimer. Opinions expressed in this presentation are from the author(s) and are not to be construed as official or as views of the U.S. Department of Commerce. The identification of any commercial product or trade names in this presentation does not imply endorsement of recommendation by NIST, nor is it intended to imply that the material or equipment identified are necessarily the best available for the purpose.

Disclaimer. Some external-source images and cliparts were included/adapted in this presentation with the expectation of such use constituting licensed and/or fair use.
List of slides

1. Randomness Beacons as Enablers of Public Auditability
2. Outline
3. Outline 1
4. Some concepts in this presentation
5. A Randomness Beacon
6. An example/conceivable application
7. Architecture of the Beacon service
8. NIST project: Interoperable Randomness Beacons
9. Outline 2
10. Some concepts useful in this talk
11. A pulse (simplified example)
12. The two “rands” in a pulse
13. Timing for generation and release
14. Fetching pulses
15. A possible diagram of pulse generation
16. Outline 3
17. Using Beacon randomness
18. Do you need to trust the Beacon?
19. Some Beacons in development
20. Some conceivable applications
21. Use case: public auditability with privacy
22. Use case: randomized clinical trials
23. Outline 4
24. Concluding remarks
25. Thank you
26. List of slides