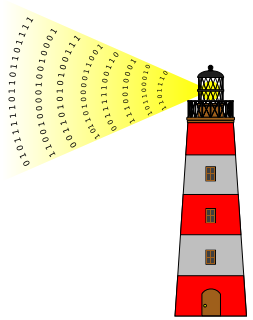
Promoting Public Randomness as a Public Good

(*Promoviendo la Aleatoriedad Pública como un Bien Público*)*

Luís Brandão and René Peralta

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



Presentation at Open Seminars IMFD
Instituto Milenio — Fundamentos de los Datos
October 4, 2019 @ Santiago, Chile

* The original was presented in Spanish (Castellano) per preference of the host. This slide deck is the English version. Some slides are based on previous presentations (ACS'19, ICMC'19).

Translated, with minor corrections, on November 18, 2019.

Outline

1. Introduction — NIST and the Interoperable Randomness Beacons project
2. Randomness Beacons — Format
3. Randomness Beacons — Operations
4. Randomness Beacons — Use
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

- ~ 6–7×10^3 workers
- Five laboratories and two centers
- Laboratories → Divisions → Groups → Projects
- Standards, research and applications

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

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.

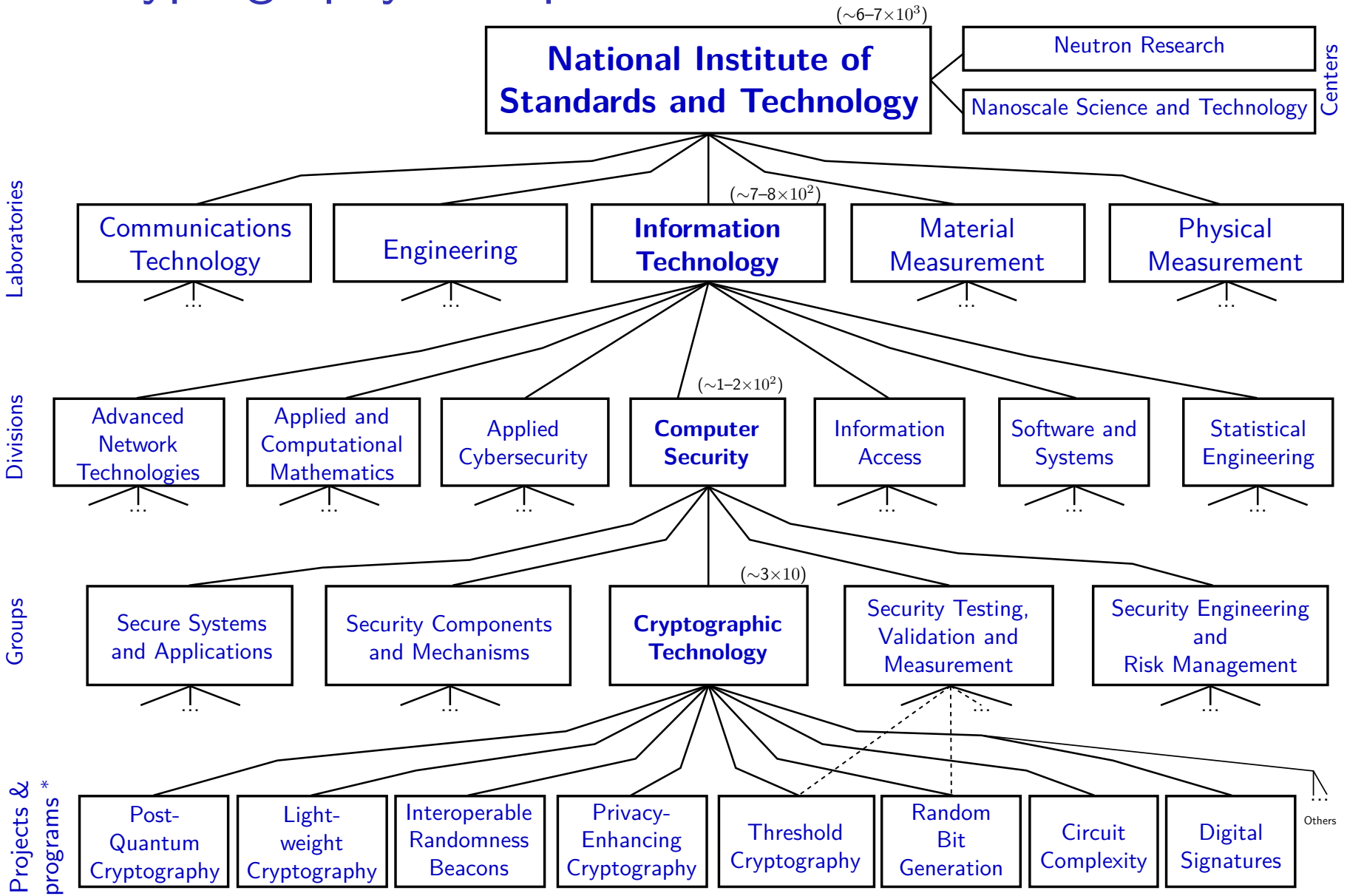
→ **Computer Security Division (CSD):** Cryptographic Technology; Secure Systems and Applications; Security Components and Mechanisms; Security Engineering and Risk Management; Security Testing, Validation and Measurement.

→ **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.

The Cryptography Group at NIST



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

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

Public Randomness as a Public Good

Public Good

Randomness

Public Randomness as a Public Good

Public Good

- [Wikipedia] "individuals cannot be excluded from use, [and] use by one individual does not reduce availability to others."

Randomness

Public Randomness as a Public Good

Public Good

- [Wikipedia] "individuals cannot be excluded from use, [and] use by one individual does not reduce availability to others."

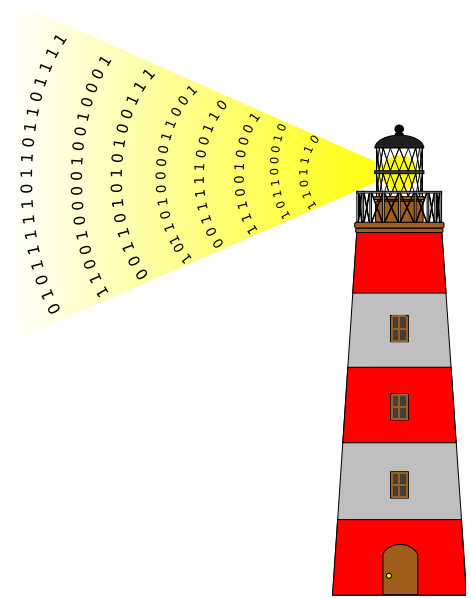
Randomness

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

A Randomness Beacon

A Randomness Beacon

A service that produces timed outputs of fresh public randomness

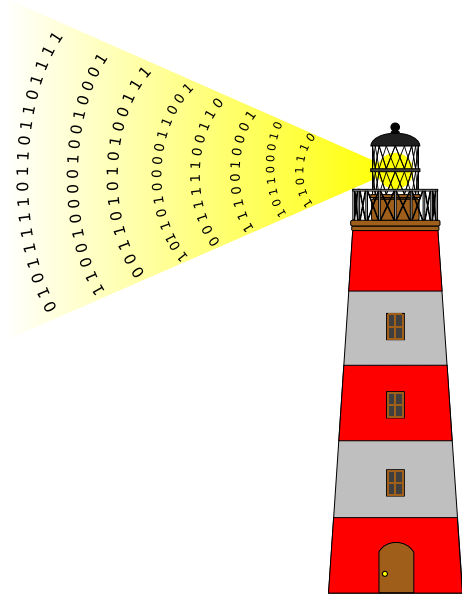


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

At a high level:

- Periodically *pulsates* randomness

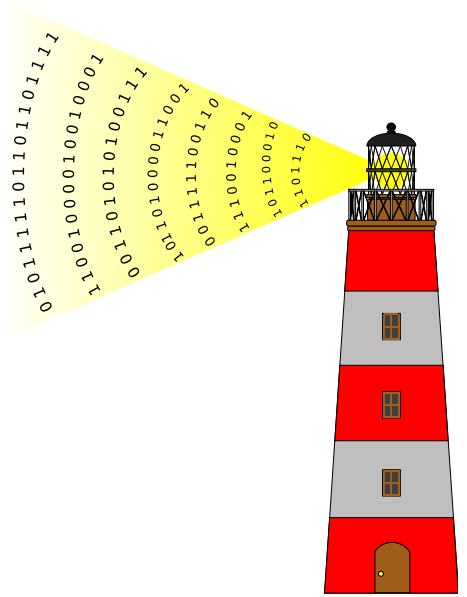


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

At a high level:

- Periodically *pulsates* randomness
- Each pulse has a fresh 512-bit random string

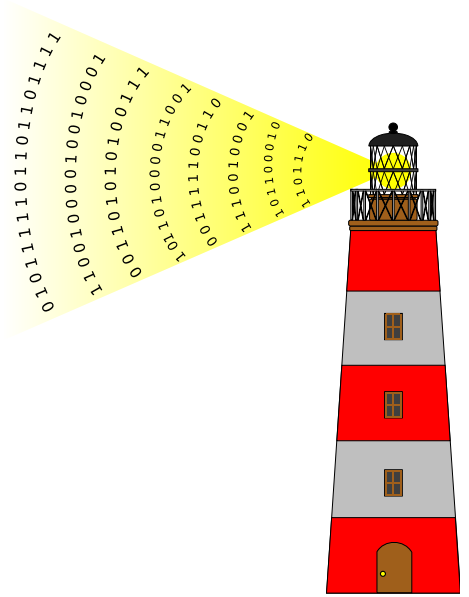


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

At a high level:

- Periodically *pulsates* randomness
- Each pulse has a fresh 512-bit random string
- Each pulse is indexed, time-stamped and signed

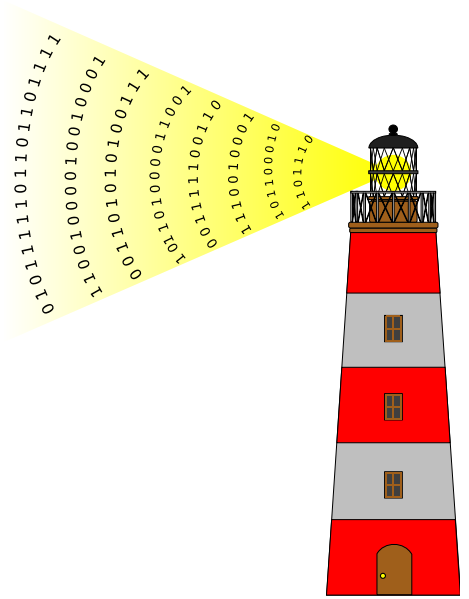


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

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

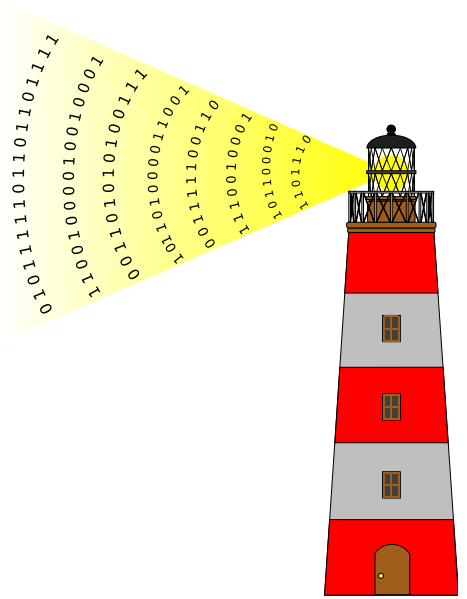


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

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

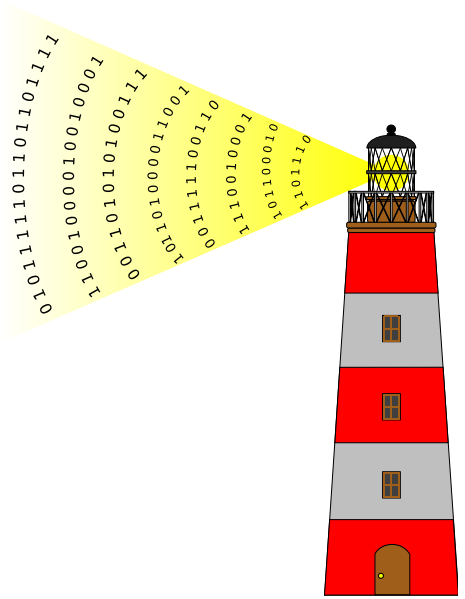


A Randomness Beacon

A service that produces timed outputs of fresh public randomness

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



Uses

- public auditability of randomized processes
- externally delegated sampling
- time-framing an event
- ...

An application in Chile

- The "Contraloría General de la República" (Comptroller) selects, at random, public officials for financial audit.
- It is natural that a selected person asks how the selection was made.
- The University of Chile is developing an application that will enable selections based on public randomness.
- In testing phase at the Comptroller https://random.uchile.cl/projects/contraloria/

Security aspects

- Is it possible to influence the randomness service to select, or not select, a particular official?
- Is it possible to attack the randomness server to know in advance which officials will be selected?

Security aspects

- Is it possible to influence the randomness service to select, or not select, a particular official?
- Is it possible to attack the randomness server to know in advance which officials will be selected?
- What interests are at stake? What resources does an adversary have?

Security aspects

- Is it possible to influence the randomness service to select, or not select, a particular official?
- Is it possible to attack the randomness server to know in advance which officials will be selected?
- What interests are at stake? What resources does an adversary have?
- The Dual_EC_DRBG case.

Security aspects

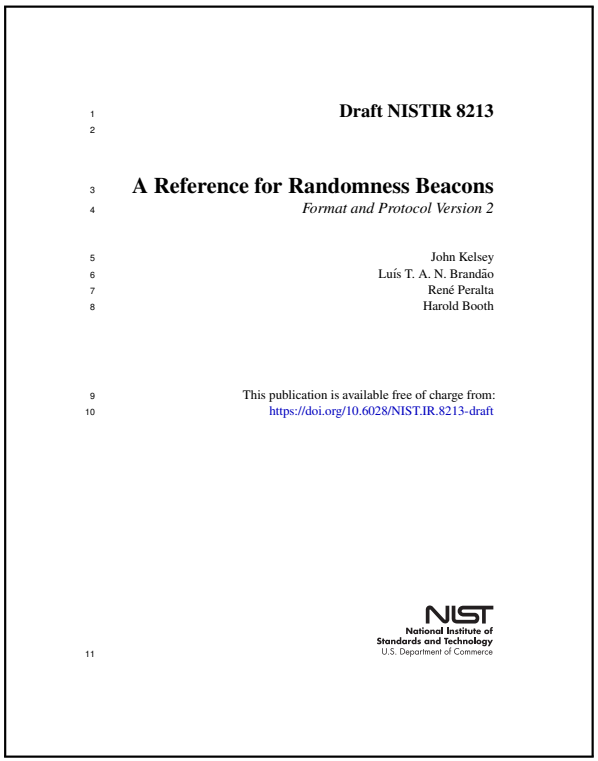
- Is it possible to influence the randomness service to select, or not select, a particular official?
- Is it possible to attack the randomness server to know in advance which officials will be selected?
- What interests are at stake? What resources does an adversary have?
- The Dual_EC_DRBG case.
- The case of the poisoned grapes.

Outline 2

NISTIR 8213 Publication

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

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



Draft NISTIR 8213

A Reference for Randomness Beacons

Format and Protocol Version 2

John Kelsey
Luís T. A. N. Brandão
René Peralta
Harold Booth

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

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

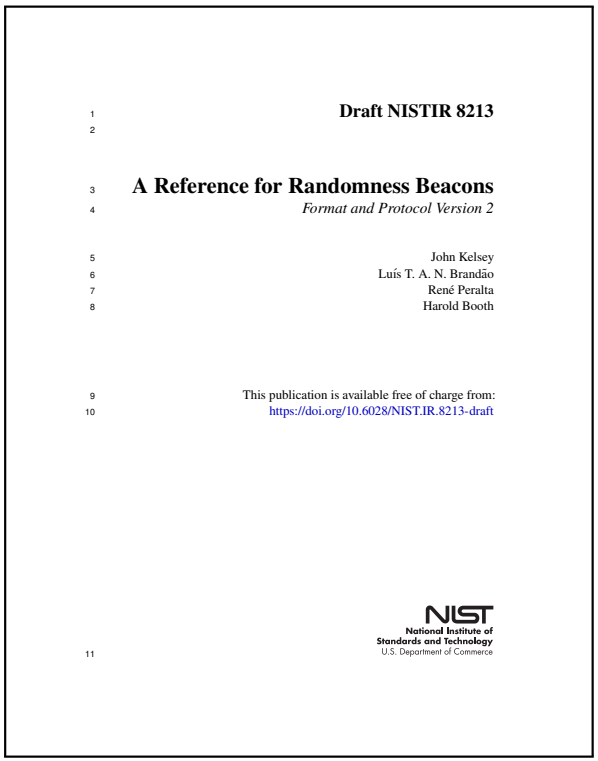
NISTIR 8213 Publication

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

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

Some topics in the report:

- format for pulses
- protocol for beacon operations
- usage of beacon randomness
- security considerations



Draft NISTIR 8213

A Reference for Randomness Beacons

Format and Protocol Version 2

John Kelsey
Luís T. A. N. Brandão
René Peralta
Harold Booth

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

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

NISTIR 8213 Publication

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

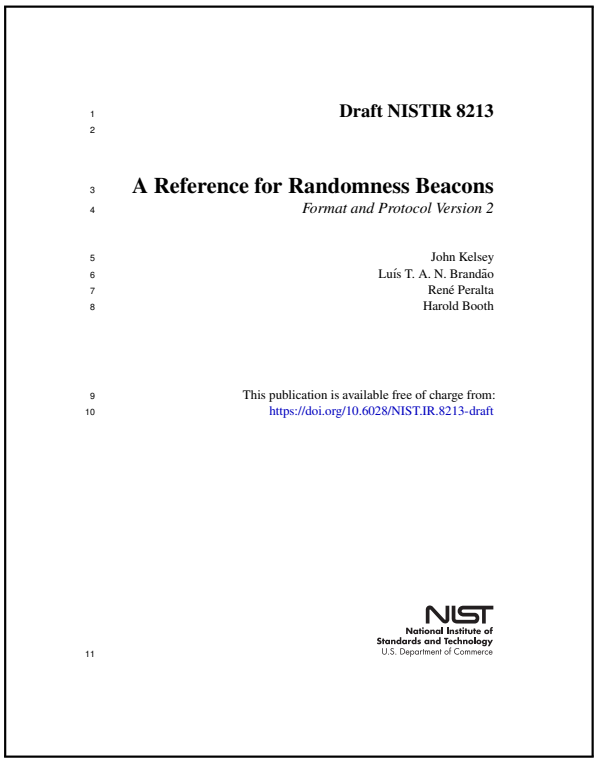
https://doi.org/10.6028/NIST.IR.8213-draft

Some topics in the report:

- format for pulses
- protocol for beacon operations
- usage of beacon randomness
- security considerations

Two goals in this presentation:

- Provide an overview of the new reference
- Motivate new implementations of public randomness and new applications for them



Draft NISTIR 8213

A Reference for Randomness Beacons

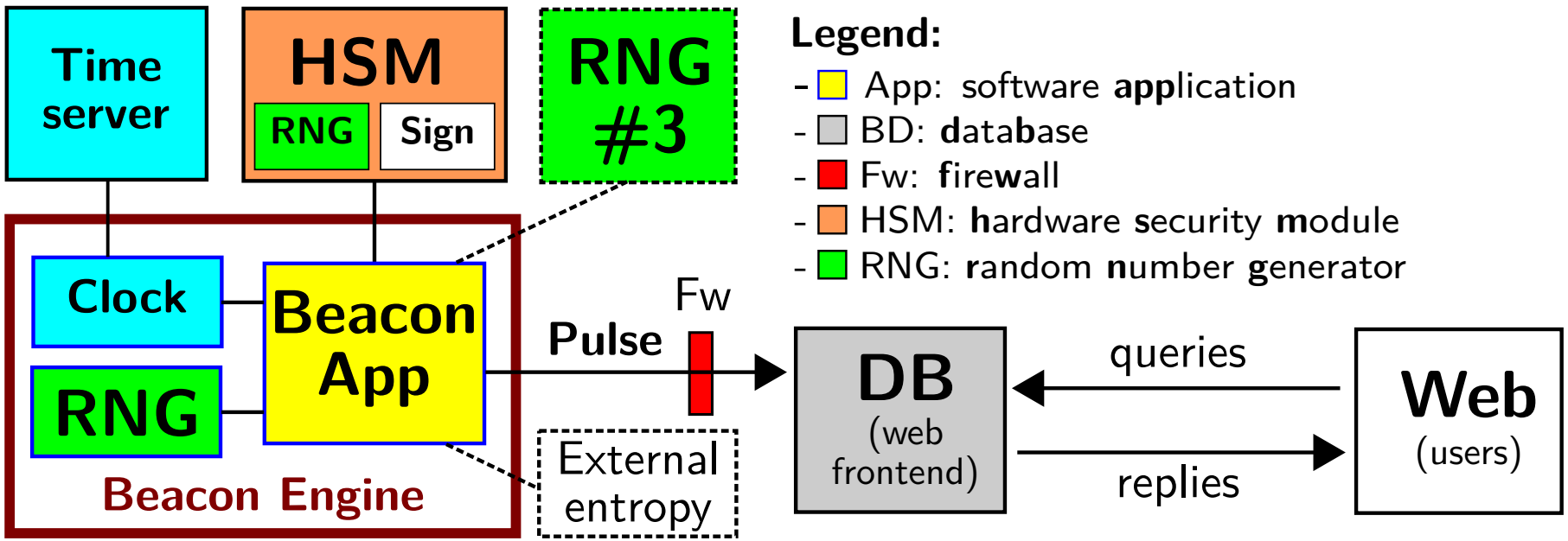
Format and Protocol Version 2

John Kelsey
Luís T. A. N. Brandão
René Peralta
Harold Booth

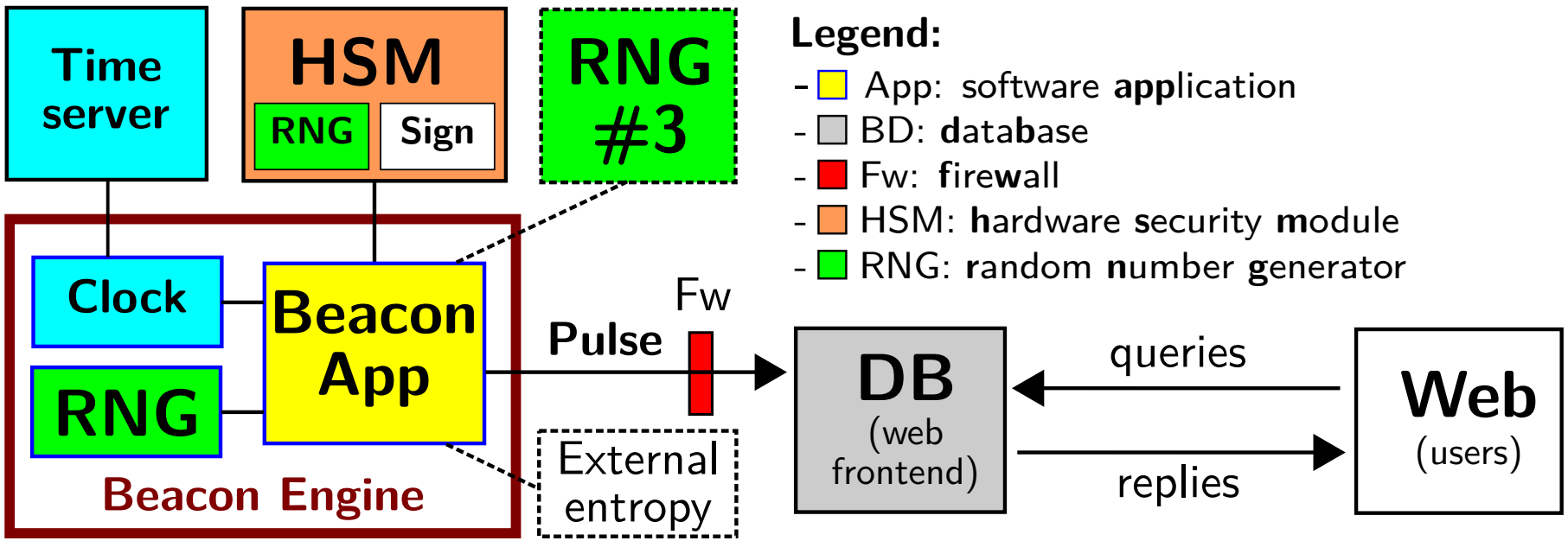
This publication is available free of charge from:
https://doi.org/10.6028/NIST.IR.8213-draft

NIST
National Institute of Standards and Technology
U.S. Department of Commerce

Architecture of the Beacon service



Architecture of the Beacon service

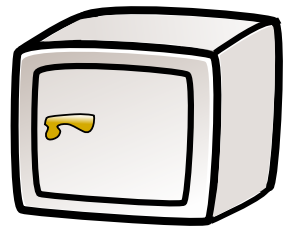


But, what exactly is a *pulse*? where does its randomness come from?, ...

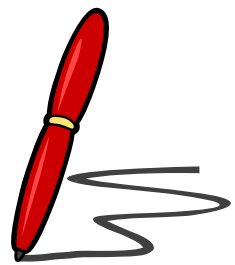
Some concepts useful in this talk

- **Hash:**

- **Commitment:**



- **[Digital] Signature:**



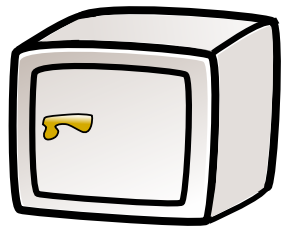
Some concepts useful in this talk

- **Hash:**

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

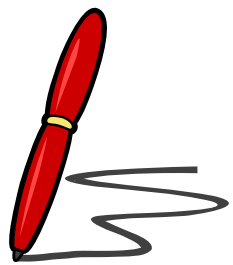
- **Commitment:**

  — 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] outPrev: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
- Useful for combining beacons

randOut (output value):

The two "rands" in a pulse

randLocal (local random value):

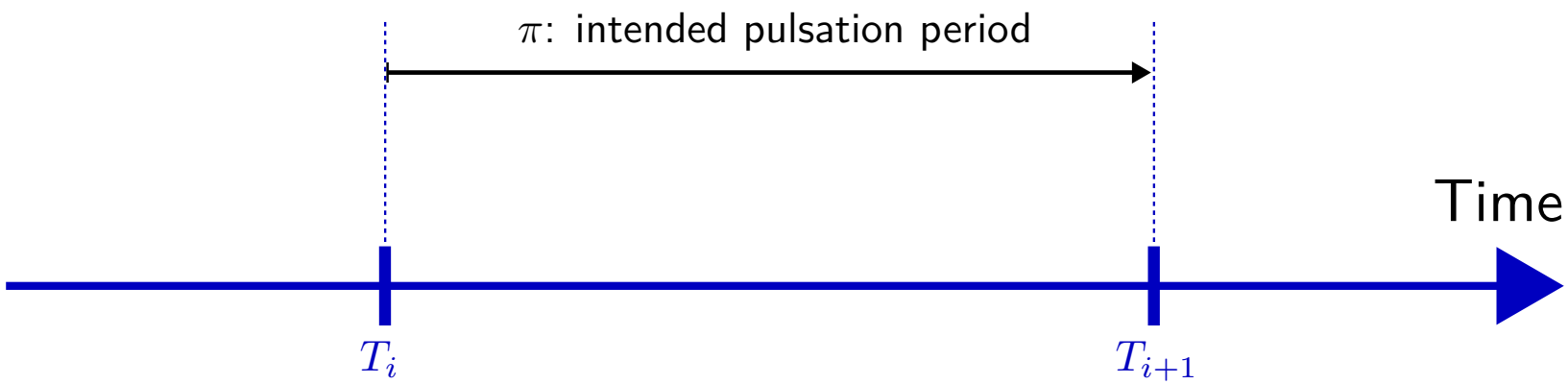
- Hash of randomness produced by ≥ 2 RNGs
- **Pre-committed** 1 minute in advance of release
- Useful for combining beacons

randOut (output value):

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

Outline 3

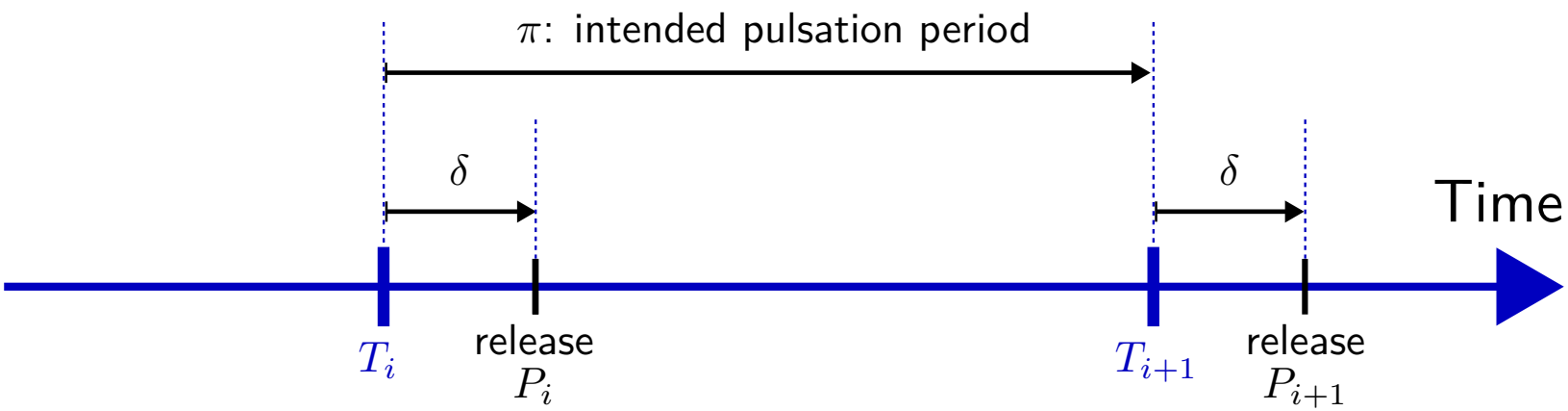
Timing for generation and release



Timing for generation and release

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

$\Big\}$ $\Rightarrow$ **Unpredictability**

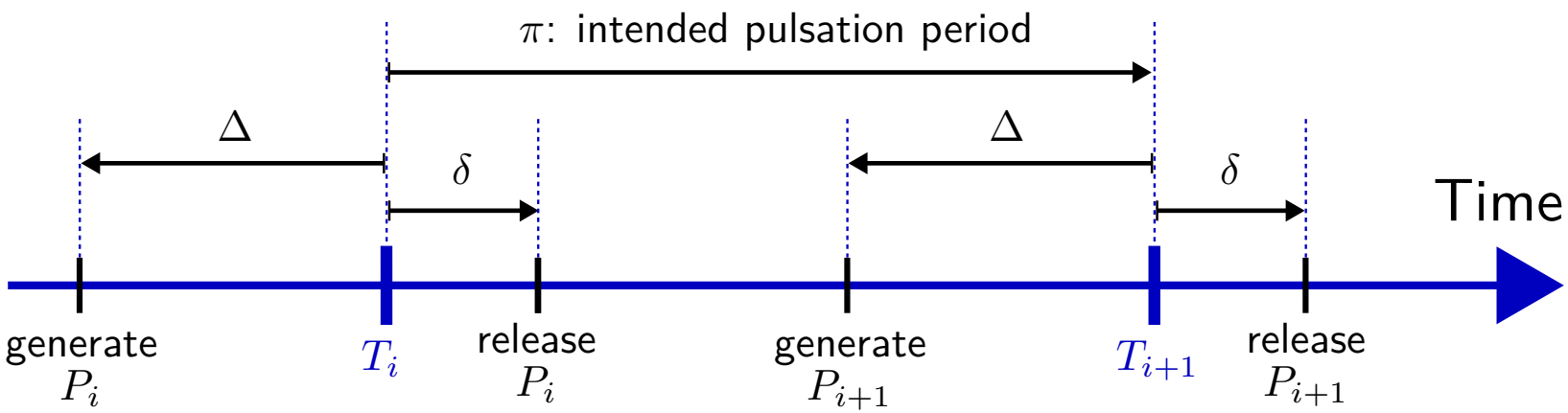


Timing for generation and release

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

$\Big\}$ $\Rightarrow$ **Unpredictability**

3. No advanced generation (small Δ) $\Rightarrow$ **Freshness**

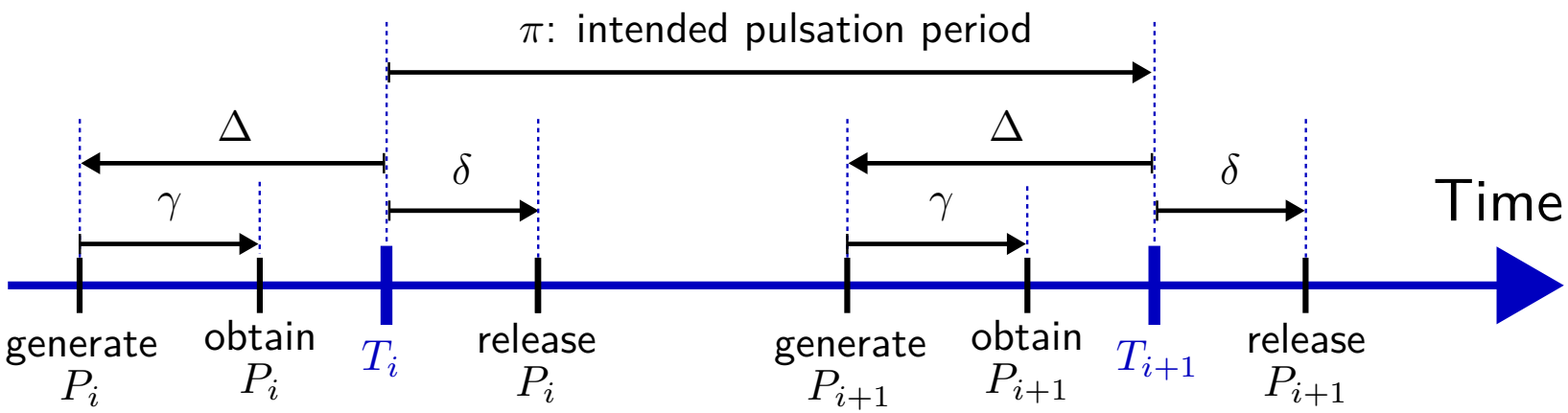


Timing for generation and release

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

$\Big\}$ $\Rightarrow$ **Unpredictability**

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

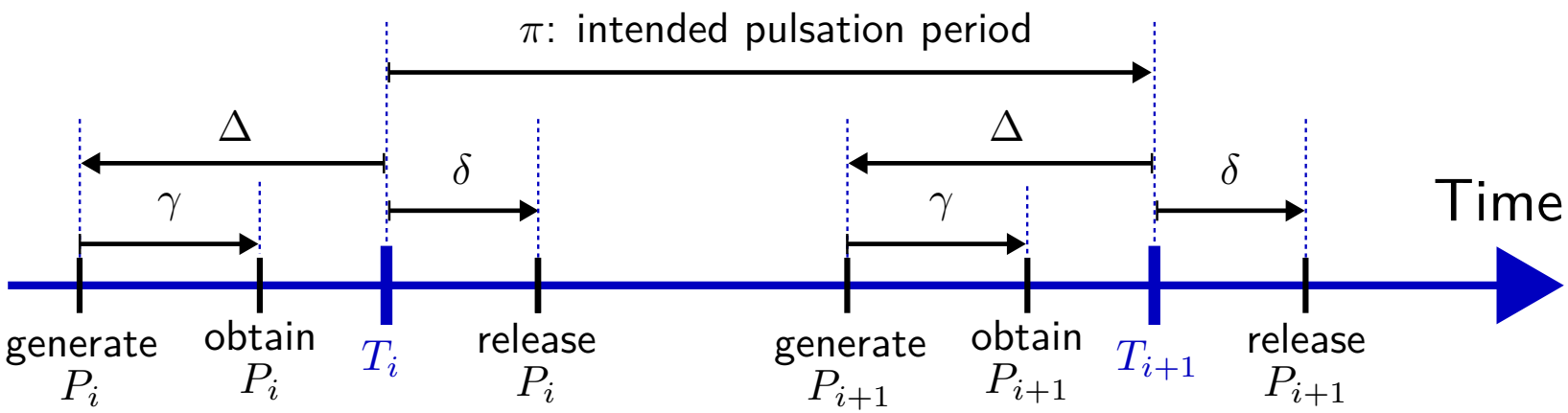


Timing for generation and release

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

$\Big\}$ $\Rightarrow$ **Unpredictability**

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

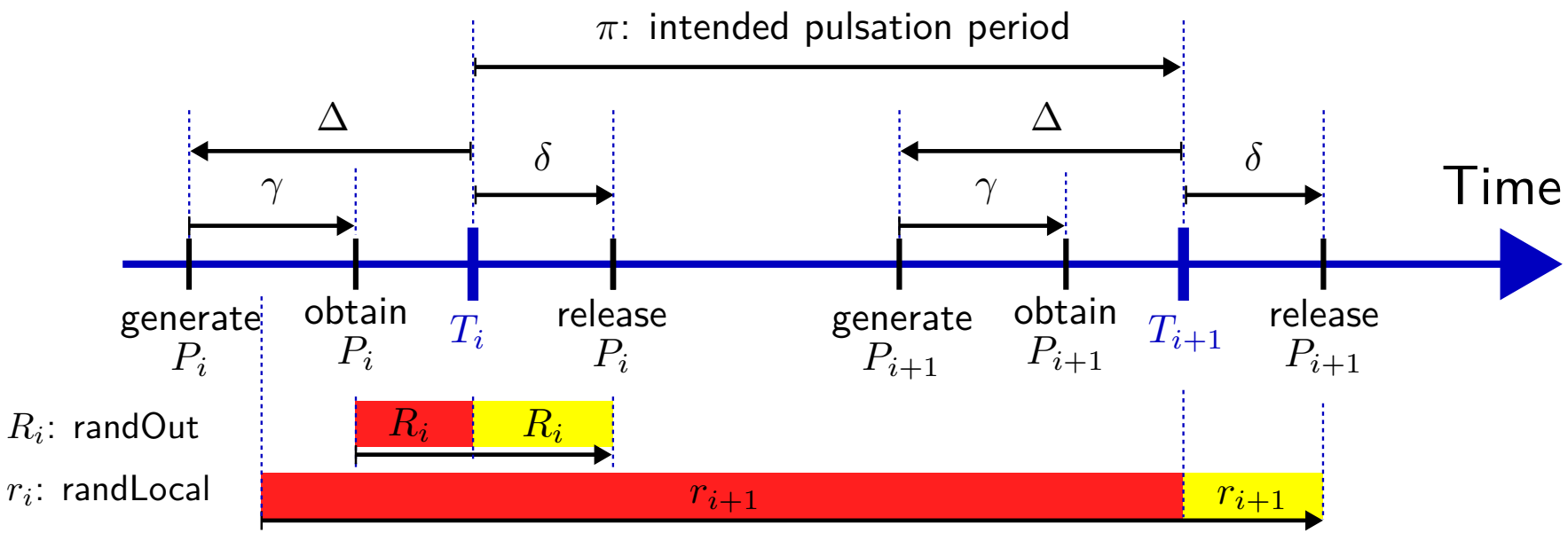


Timing for generation and release

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

$\Big\}$ $\Rightarrow$ **Unpredictability**

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

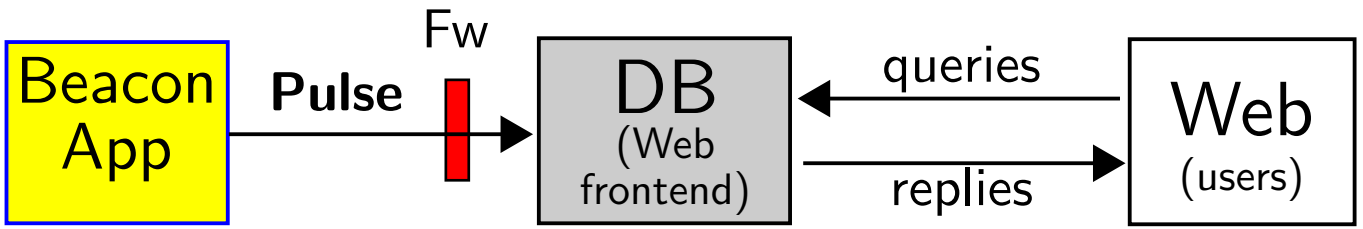


(The actual requirements specify allowed intervals for δ and Δ)

Fetching pulses

Fetching pulses

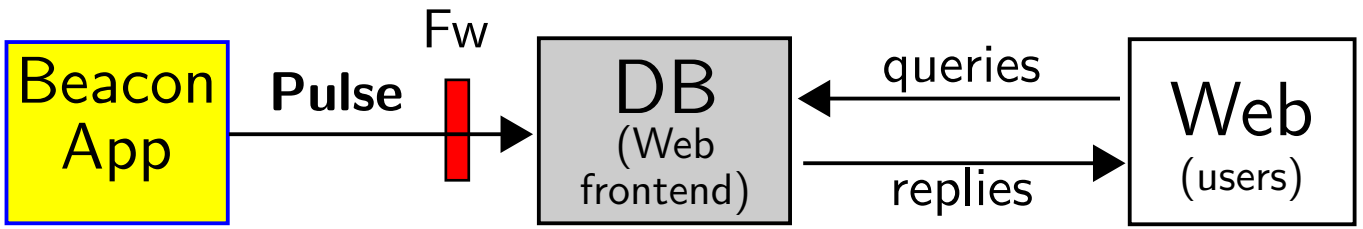
Beacon App: a pulse release means sending it to the database



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



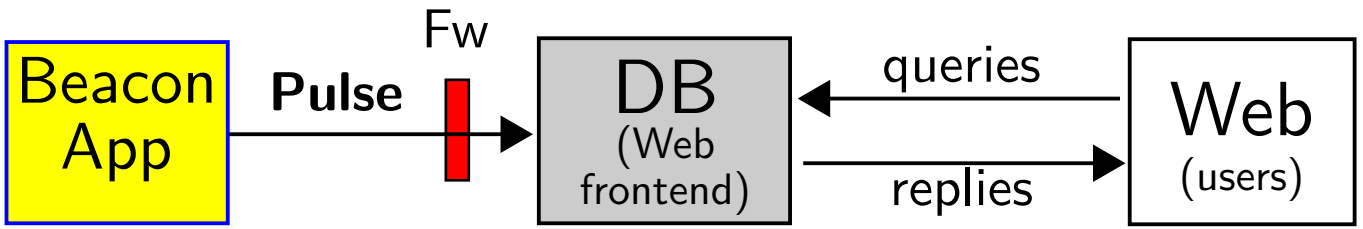
Legend: App: **app**lication; DB: **d**ata**b**ase; Fw: **f**ire**w**all.

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:

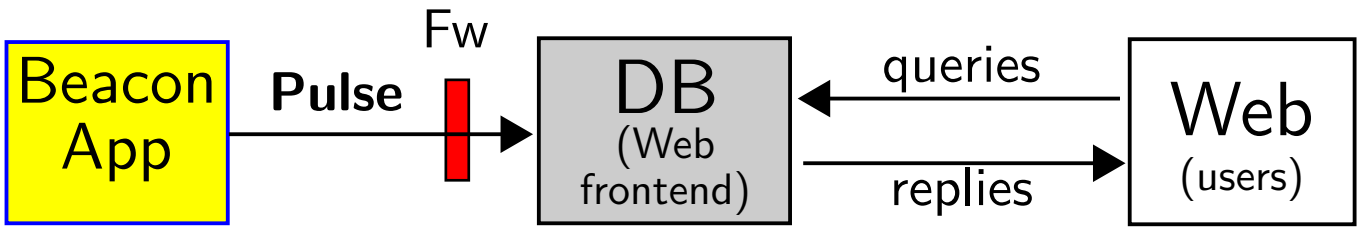
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

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:

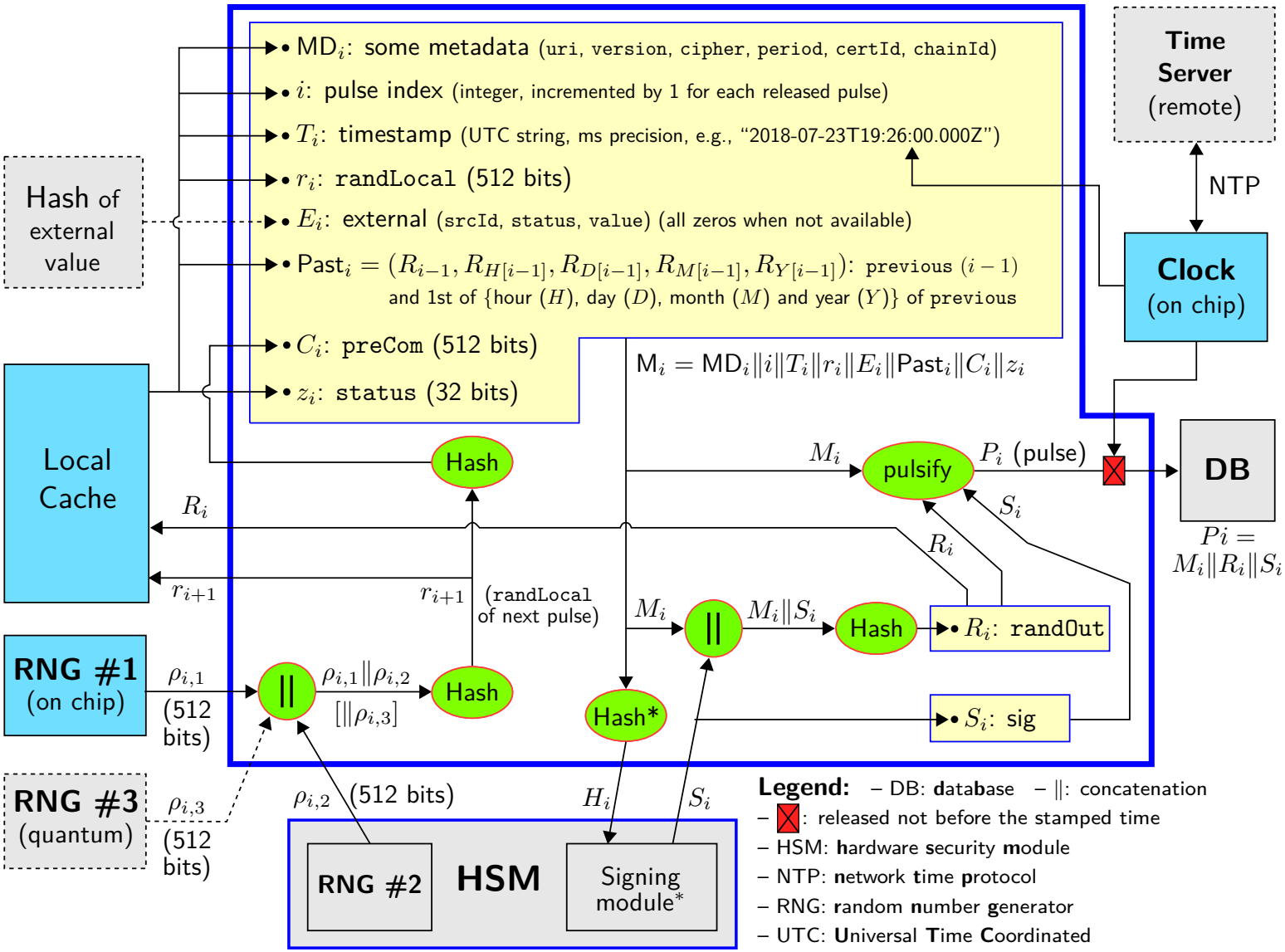
(URI = **u**niform **r**esource **i**dentifier; URL = **u**niform **r**esource **l**ocator)

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 4

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 $R = \texttt{randOut}[t]$ (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 $R = \texttt{randOut}[t]$ (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 Contraloria), 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

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 and 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

Some conceivable applications

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

Example applications:

- Select random test and 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

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 and 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

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

Outline 5

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**
- **Planned:**
  - Complementary analysis and guidance
  - Improvements based on feedback

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**
- **Planned:**
  - Complementary analysis and guidance
  - Improvements based on feedback
- **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

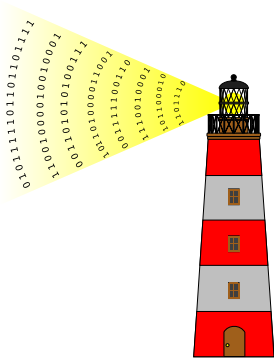
Thank you

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

Promoting Public Randomness as a Public Good

(Promoviendo la Aleatoriedad Pública como un Bien Público)

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



Presentation at Open Seminars IMFD

October 4, 2019 @ Santiago, Chile

List of slides