A Yarn of Randomness

The CU Randomness Beacon and the Twine protocol

Jasper Palfree - University of Colorado; NIST

Presented at the NIST Crypto Reading Club
2024-02-21
Randomness Beacon

Source of public randomness

Emits bits at regular intervals (usually)

Can be verified to be unbiased and unpredictable

Maintains an immutable record of past values
Useful for enabling unbiased choices needed in high-stakes decisions
Why build another beacon?

There are already randomness beacons

- NIST Randomness Beacon Gaithersburg
- Singapore Randomness Beacon (in development)
- Random UChile
- DRAND (Cloudflare)
2003
The Bell Experiment confirms that nature, as described by quantum mechanics, behaves in a fundamentally non-deterministic way.

NIST Team Proves 'Spooky Action at a Distance' is Really Real

November 10, 2015

BOULDER, Colo.—Einstein was wrong about at least one thing: There are, in fact, "spooky actions at a distance," as now proven by researchers at the National Institute of Standards and Technology (NIST).

Einstein used that term to refer to quantum mechanics, which describes the curious behavior of the smallest particles of matter and light. He was referring, specifically, to entanglement, the idea that two physically separated particles can have correlated properties, with values that are uncertain until they are measured. Einstein was dubious, and until now, researchers have been unable to support it with near-total confidence.

As described in a paper posted online and published in Physical Review Letters (PRL), researchers from NIST and several other
Bell Experiment
DIRNG (Device-Independent Random Number Generation) uses Bell Experiment data to extract entropy from that non-deterministic part of nature.

This is great for generating truly random numbers.

Experimental Low-Latency Device-Independent Quantum Randomness

Yanbao Zhang,1,* Lynden K. Shalm,2,* Joshua C. Bienfang,3 Martin J. Stevens,2 Michael D. Mazurek,2 Sae Woo Nam,2 Carlos Abellán,4,1 Waldimar Amaya,4,1 Morgan W. Mitchell,4,5 Honghao Fu,6 Carl A. Miller,6,3 Alan Mink,3,7 and Emanuel Knill2,8

1NTT Basic Research Laboratories and NTT Research Center for Theoretical Quantum Physics, NTT Corporation, 5-1 Morinosato-Wakamiya, Atsugi, Kanagawa 243-0198, Japan
2National Institute of Standards and Technology, Boulder, Colorado 80305, USA
3National Institute of Standards and Technology, Gaithersburg, MD 20899, USA
4ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain
5ICREA-Institució Catalana de Recerca i Estudis Avançats, 08010 Barcelona, Spain
6Department of Computer Science, Institute for Advanced Computer Studies, and Joint Center for Quantum Information and Computer Science, University of Maryland, College Park, MD 20742, USA
7Thesis Research, La Jolla, CA 92037, USA
8Center for Theory of Quantum Matter, University of Colorado, Boulder, Colorado 80309, USA

Applications of randomness such as private key generation and public randomness beacons require small blocks of certified random bits on demand. Device-independent quantum random number generators can produce such random bits, but existing quantum-proof protocols and loophole-free implementations suffer from high latency, requiring many hours to produce any random bits. We demonstrate device-independent quantum randomness generation from a loophole-free Bell test with a more efficient quantum-proof protocol, obtaining multiple blocks of 512 random bits with an average experiment time of less than 5 min per block and with a certified error bounded by $2^{-54} \approx 5.42 \times 10^{-20}$. 
DIRNG (Device-Independent Random Number Generation)

Raw Data → Certificate of entropy → Extraction → 512 bits certified randomness

Random Seed

512 bits certified randomness
Benefits

With other RNG methods it's very difficult to verify the process is truly random.

With DIRNG the output is certified to have a certain amount of entropy, and that entropy must have come from a non-deterministic process.

...regardless of the experimental setup.
Building a randomness beacon

NIST wrote instructions

A Reference for Randomness Beacons

Draft NISTIR 8213

Format and Protocol Version 2

John Kelsey
Luís T. A. N. Brandão
René Peralta
Harold Booth
Building a randomness beacon

In a nutshell...

A hash-chain of digitally signed pulses
Building a randomness beacon

In a nutshell...
Building a randomness beacon

In a nutshell...
Building a randomness beacon

In a nutshell...

private randomness

# (\textcolor{cyan}{+} \textcolor{cyan}{+} \textcolor{red}{+})
Building a randomness beacon

Challenge 2:

DIRNG extracts randomness from raw bell data using a random seed which influences the output randomness.

Need a way to prove that the seed was chosen by a neutral party.

Need a way to prove that we pre-committed to using a value that we didn't know at the time we committed to it.
Building a randomness beacon

ONE DOES NOT SIMPLY
BUILD A RANDOMNESS BEACON
One protocol to rule them all?

The Twine Protocol

Three separate processes = Three separate chains

- Raw Data
- Certificate of entropy
- Random Seed

Extraction

512 bits certified randomness
One protocol to rule them all?

The Twine Protocol

Three separate processes = Three separate chains

Pulses are generic containers.

Each chain can have its own sub-protocol for its own use-case.

Twine pulse

```json
{
  ...
  specification: "twine/1.0.x/nist-rng/1.0.x"
  payload: { arbitrary data }
}
```
Cross-chain hash linking

The Twine Protocol

Pulses include hashes of pulses on other chains

Serves two purposes:

1. Time bound on the output hash. Blue pulse hash can not be known until after purple pulse is published and vice versa.
Cross-chain hash linking

The Twine Protocol

Pulses include hashes of pulses on other chains

Serves two purposes:

1. Time bound on the output hash. Blue pulse hash can not be known until after purple pulse is published and vice versa.
2. Preserves the history of the external chain. Blue chain’s past pulses cannot be rewritten without purple chain knowing.
Stitching pulses of different chains together using hash-linking provides both tight time bounds and data integrity.
A stitch in Twine saves time
A different way to think about time-stamping
This creates a DAG (Directed Acyclic Graph)
Pathfinding implies causal ordering

Past

Future
But in general a DAG only implies partial ordering
Compare it with spacetime
Stitching together chains creates a "hash-time"

Information propagates at the "hash-speed"
This is more faithfully reflects the true (relativistic) nature of time.
It takes three to tangle

CU Chain carries out the DIRNG protocol
NIST Chain records experiment events
DRAND wrapper Chain acts as the seed chain

Communication happens through Twine.

CU requests data. NIST provides data privately.

CU publishes precommitment pulse with hash of raw data, seed length, seed source (chain).

Once seed becomes available, randomness is extracted and published along with the raw data.
The CURBy Tangle
Live demo?
CURBy (CU Randomness Beacon)

What is CURBy?
CURBy is an acronym for the CU Randomness Beacon (why? because we can!). CURBy is a public service free for anyone to use. Its goal is to provide a new way to generate random values extracted from the quantum vacuum in a way that can be verified by anyone.

You can browse the past and present random values below:

<table>
<thead>
<tr>
<th>CURBy</th>
<th>CURBy-Quantum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewing Pulse ID: 12345678909876543210</td>
<td></td>
</tr>
<tr>
<td>Mixing in with 5 other chains</td>
<td></td>
</tr>
<tr>
<td>c9c368456a588b8f97a2a94141dc5b53c53b1e57a9a0798515e2a52815f</td>
<td></td>
</tr>
<tr>
<td>a6b6eb7e8b48b5a250f76e613a8b5b708ab6</td>
<td></td>
</tr>
<tr>
<td>released: 14:38:08 on Thursday, February 15, 2024</td>
<td>pulse: 161507 / 161507</td>
</tr>
</tbody>
</table>

Why create a randomness beacon?
Often randomness is thought of as something you want kept hidden, such as when generating passwords or cryptographic keys. But broadcasting random values that everyone can agree on is incredibly useful for making decisions without human bias.

Consider the situation of deciding which ballot boxes should be audited in an election. Allowing humans, especially those in power, to make this decision leaves our democratic process more susceptible to corruption. Instead, guaranteeing that the decision is random and proving it to be outside of the control of any organization helps maintain the fairness of elections. This is what CURBy can do; prove that a random choice was unpredictable and outside of anyone's control.

What makes CURBy special?
Publicly agreed upon randomness is not a new idea. There have been many randomness beacons before CURBy, and CURBy's implementation has taken inspiration from its predecessors such as The League of Entropy's Distributed Randomness Beacon and especially The NIST Randomness Beacon.

What makes CURBy special is that it addresses two main pitfalls found in past beacons.

Challenge #1: Unpredictability vs. Randomness
The League of Entropy's beacon involves a network of computers that collectively generate pseudo-random values at regular intervals. If the nodes in the network are operated independently up to some threshold, the values are unpredictable, but they are still deterministic; that is, predetermined from the start. This means that, while perhaps unlikely, it is possible for a large enough subset of nodes to coordinate and predict future values.

Challenge #2: Trust vs. Verifiability
Here Ralphie, CU's mascot, is doing a random walk based on the latest random pulse.
CURBy Network (actual data)
Features that make Twine flexible

- Generic payloads and sub-protocols.
Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)

CID INFO
QmW7z2K2znPOexzdFOz7HGiTUCQG7Hh3xyqy94czyJQC
base58btc - cidv0 - dag-pb - sha2-256~256~6A858049CB305292AA7084FAB18A5...
BASE - VERSION - CODEC - MULTIHASH

MULTIHASH
0x12326A858049CB305292AA7084FAB18A5DC6
6FC8B3594167FAE4CC4366384AF36595
HASH DIGEST

0x12 = sha2-256
0x32 = 256 bits
Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)
- Chain data structures associate chains with CIDs

CID(bafyriqa5k2d3t3r774geicu...)

Chain Meta Data

{ public key
 structural meta data
 source meta data
}
Features that make Twine flexible

- Generic payloads and sub-protocols.
- Content addressing and CIDs (hashes are the identifier)
- Chain data structures associate chains with CIDs
- Flexible storage options
Possible futures

"Cat's Cradle" is more fun with friends
Thanks

Here are some key takeaways

- Twine is still in development but very fleshed out and easy to develop with
- CURBy is the first example use-case but hopefully the first of many
- Welcome interest for testing it out and solving outstanding questions
  jasper.palfree@colorado.edu