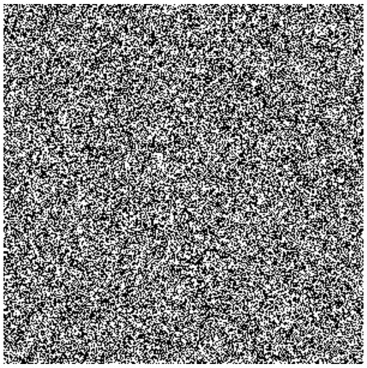
Issues
Contribute
Blog
About Us

Search

Shop



ISSUE NUMBER SIX: THE TOTAL ARCHIVE

Exhibit: The Entropy Archives

What does a perfectly random archive look like? **Finn Brunton** explains.

https://beacon.nist.gov/home

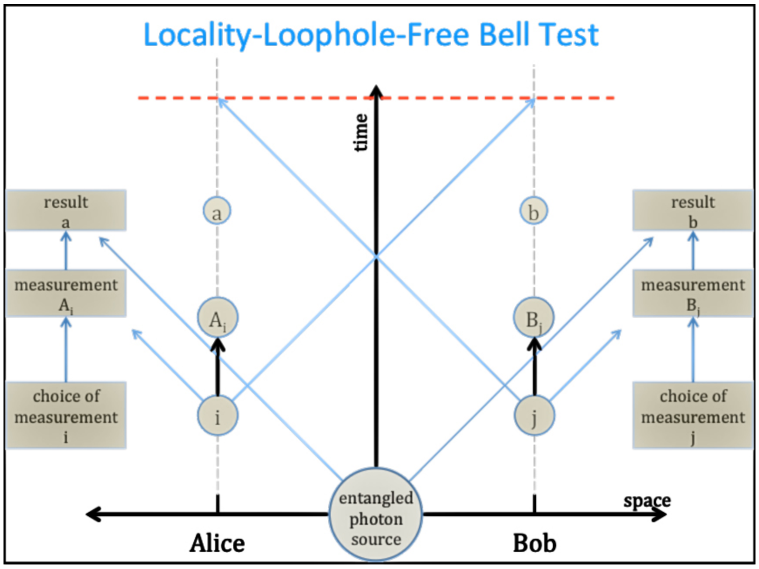
The National Institute for Standards and Technology (NIST) maintains an archive of the generic and the default called the Standard Reference Materials. It is an archive of a very specialized sort: not one of particular objects, but instead of the most generic objects possible, against which others can be benchmarked, simultaneously completely typical and as specific as exacting measurement and engineering can make them. NIST has cigarettes for testing the ignition resistance of furniture, waterway sediment, crude oil, slurried spinach, argillaceous limestone, and reference peanut butter. Starting on September 5, 2013, a bit before noon, they also began producing standard random objects at a rate of one per minute: strings of 512 bits of entropy, broadcast every 60 seconds. (They call it a "public randomness service.") The first one starts like this: "17070B49D …".

This "public randomness beacon" starts with the combination of two independent pieces of hardware that generate random numbers. The resulting 512-bit number is an excellent source of randomness, which is then combined with all the data pertinent to that particular value: the version number, the timestamp of its creation, output frequency, a status code, and—most significant, for the question of archivesthe value of the *previous* output, the most recent random broadcast. This collection of data is then "hashed," or run through a function that takes data of any length and produces data of fixed length so that any change to the original data changes the hash output. You dump the data in the hopper, and get a string of characters ("63C4B71D51…") that preserves the original randomness while also being trivial to verify that it corresponds to its time and status information. This string is then signed with NIST's private key, a cryptographic tool for proving that NIST in fact sent it; that collection of data is hashed again, and at last you have the output value.

The result is an abstract kind of archive with a set of powerful properties, a set of characters that contains no information in a mathematical sense—each character is unpredictable based on past

- Zebras, Blanks and Blobs
- **Exhibit: The Entropy Archives**

activity, and tells you nothing about the next character to come—but that can also verify that it is the product of NIST, sent by those who claimed to send it.



Let's say you need to randomly recount ballots from some districts to verify the integrity of a vote. How can everyone be sure you're using actually random numbers to choose the districts? If you get to select the numbers, you could rig the election. So you use the output of the public randomness beacon. What if you and your co-conspirators fake the output of the beacon? Maybe you can generate fake "random" characters and get access to NIST's private key to sign them, so you know in advance what the random draw will be. Now think about the reliance on random quality assurance checks in manufacturing everything from cars to pharmaceuticals, in conducting medical screenings, as components of stock market trading strategies, and even in military decisions—the safest evasive maneuver, all other things being equal, is one your opponent can't predict because even you don't know it in advance—and the importance of having *reliable* unreliable numbers becomes clear.

This is why the *archive* of entropy is so vital: because every new random string is hashed with the previous one, it is easy to verify the output of a hash, and the output is extremely difficult to predict in advance. This means you can generate a fake "random" string, and steal NIST's key to send it out, but anyone can check whether it incorporates the randomness of the previous broadcast. Which it won't, unless you faked that one too, but that in turn needs to incorporate the broadcast before it, link by link, two years back, a minute at a time. In other words, to produce randomness everyone can trust—randomness that reveals no information about future randomness, a perfectly level probability landscape—it has to be part of an archive of a very special kind, a timeline that verifies nothing but its own integrity.