TRUST, AND PUBLIC ENTROPY: A UNICORN HUNT

Arjen K. Lenstra and Benjamin Wesolowski
WHAT IS PUBLIC RANDOMNESS

And what is it good for?
ELEMENTARY EXAMPLES

National lotteries

Sporting event draws

Tie breaking in elections

Totally based on randomness (presumably), and huge amounts of money or power at stake
A TOOL FOR DEMOCRACY

First known democracy in the world, in Athens: legislative and judicial power distributed to assemblies of randomly selected citizens

Require a secure random sampling procedure, that every sceptical citizen can trust and verify
M. O. Rabin.

Transaction protection by beacons

Introduces the notion of **random beacon**: A random beacon is an online service broadcasting (allegedly) unpredictable random numbers at regular intervals (say, every minute).

...00111100010101

**random beacon = public stream of random numbers**
TRANSACTION PROTECTION BY BEACONS

A few applications of trustworthy public randomness:

➤ **transaction protocols**: fair contract signing, confidential disclosure, mail certification

➤ **choice of standard parameters**: standard elliptic curves, constants in S-Boxes or round constants in hash algorithms…

➤ random challenges for **cryptographic elections**

➤ **smart contracts** in crypto-currencies: secure lotteries, non-interactive cut-and-choose protocols…

➤ **preventing selfish mining** in crypto-currencies
GENERATING PUBLIC RANDOMNESS

Can you trust someone else's entropy
THE (GOOD?) OLD WAY

a kleroterion
2600 YEARS LATER

Can the security be upgraded?…
J. Clark and U. Hengartner.

**On the use of financial data as a random beacon.**

Easy to imagine that financial exchanges could subtly adjust the prices they announce to bias the “random” output.
COMBINING LOTTERIES

seed → results of national lotteries around the world in February 2016

public deterministic procedure → published in January 2016

elliptic curve → The Million Dollar Curve

http://cryptoexperts.github.io/million-dollar-curve/, CryptoExperts
COMBINING LOTTERIES

➤ Cannot produce a regular stream of numbers like a beacon (not a problem for their application)
➤ Last draw attack
➤ Again, you have to trust some third party...

http://www.businesspundit.com/5-of-the-biggest-lottery-scandals/
THE NIST RANDOM BEACON

- 512 random bits per minute
- generated based on quantum mechanical phenomena, “true randomness”
- No proof that the numbers are properly generated can be provided
Can we get rid of the trust assumptions, in favor of computational assumptions?
The Bitcoin blockchain

00000000
transactions
64465734
The Bitcoin blockchain
The Bitcoin blockchain

00000000
transactions
36457740
hash
09436663
The Bitcoin blockchain
The Bitcoin blockchain

00000000
transactions
86797810
hash
00004339
The Bitcoin blockchain

BITCOIN ENTROPY
The Bitcoin blockchain

00000000 transactions 00004339 transactions
86797810 hash 45364536 hash
00004339 00007522
BITCOIN ENTROPY

The Bitcoin blockchain

- 00000000 transactions
  - 86797810 hash: 00004339
- 00004339 transactions
  - 45364536 hash: 00007522
- 00007522 transactions
  - 00119427 hash: 00001294
Finding such that starts with enough leading zeros is called mining, performed by miners, who get a reward when they find a valid block.
Idea: use 4339 as a random number

Protocol is decentralised, mining is costly. Should render manipulations difficult

How difficult?
Idea: use 4339 as a random number

Problem: Groups of colluding miners can bias the output

If 25% of the miners are colluding, they can bias a coin toss from probability 0.5 to 0.74!

(Antpool and F2Pool each control more than 26%)

Numbers from Cécile Pierrot and B. W., *Malleability of the blockchain’s entropy*, to be presented at ArcticCrypt Conference 2016
A random zoo: sloth, unicorn and trx.
UNICORN: UNCONTESTABLE RANDOM NUMBERS

1. **Open protocol**: anyone is able to take part in the generation process (and it is very easy)
2. **Verifiable**: anyone can verify everything went right
3. **Secure**: even if only one single participant is honest (and that can be you, thanks to 1.)
Observation: a number can be fully determined at point in time $t$, while none of its bits can be known by anyone before time $t + \Delta$, for some delay $\Delta$.
**UNICORN: UNCONTESTABLE RANDOM NUMBERS**

- **data generated at time t**
- **slow-timed hash (sloth)**
- **34560039**

_Sloth_ must be guaranteed to take time at least $\Delta$ to compute, irrespective of available parallel resources.

Trivial example: SHA-2 iterated millions of times.

Better example: _sloth_, based on square root extractions in finite fields (efficiently verifiable, with only some squarings).
UNICORN: UNCONTESTABLE RANDOM NUMBERS

➤ Latest news at time $t$, weather data, stock values, latest output of the NIST beacon
➤ Screenshot of a public online bulletin board
➤ Latest tweets containing the hashtag #unicorn

By sending a tweet at the right moment, you are guaranteed nobody knew before time $t$
At time $t$, the input of *sloth* is published, and the computation begins.
By sending a tweet at the right moment, you are guaranteed nobody knew the data generated at time $t$ before time $t + \Delta$.

$sloth$ takes time $\Delta$ to finish.

Thus, not a single bit of the data is known before $t + \Delta$.
UNICORN: UNCONTESTABLE RANDOM NUMBERS

data generated at time $t$ → slow-timed hash (sloth) → 34560039

not a single bit of is known before $t + \Delta$

+ is fixed (and public) at time $t$

= Nobody can willingly bias even a single bit of
DESIGNING A SECURE RANDOM BEACON

Guarantees and constraints
TRUSTWORTHY ENTROPY, RATHER THAN TRUSTED ENTROPY

Get rid of the trust assumption: prove to everybody that your random numbers are not manipulated
THE TRUMAN SHOW MODEL

A user of a secure beacon may trust nobody but himself

➤ lotteries are rigged
➤ Bitcoin miners are all colluding against him
➤ and with everybody else in the world but him

Yet he should still be able to verify that the output numbers are not manipulated
OPEN PUBLIC INPUT

The *unicorn* protocol needs public input, for people to make sure the data wasn’t known by anyone before $t$

We argue open public input is necessary in the Truman Show model, in order to **fix the random number in time** even for the most skeptical users
The *unicorn* protocol suffers a delay in its execution.

We also argue that in this model, there must be a delay separating the moment where the output is determined \((1)\), and the moment it can be known \((2)\).