PUBLICLY VERIFIABLE RANDOMNESS

SHAFI GOLDWASSER, SALEET KLEIN, ELCHANAN MOSSEL AND OMER TAMUZ

Abstract. When the outcome of an algorithm or an experiment depends on a random or pseudo-random input, there may exist incentives to manipulate this input to achieve desired results. We propose a public ledger mechanism by which it is made possible to publicly verify that an input is indeed random and has not been manipulated. Applications include medical experiments, lotteries, ordering of co-authors on academic papers, financial audits and arms treaty inspections.

1. Introduction

Randomness plays a crucial role in many algorithms and processes. In many applications of randomness, it is desirable to verify that what was supposed to be random is indeed random. Such applications include medical and laboratory trials, lotteries, arms treaty inspections, and financial and electoral audits. State run lotteries have traditionally addressed this problem by using public physical randomization devices, such as ball machines, to prove that their draws are random. These have been successfully manipulated in the past [7].

We propose a simple protocol based on blockchains which delivers to users randomness which is guaranteed not to have been manipulated.

1.1. Untrusted randomness.

The main objection to the use of randomness by individual parties is that there is no reason to trust that they indeed used a random source for their application, nor is it possible to exclude that they specifically chose their random input to achieve their desired results.

A natural first idea for addressing this issue is to use a publicly known *hash function* to choose a pseudo-random number. A hash function $H\colon \{0,1\}^* \to \{0,1\}^\ell$ assigns a length ℓ string of zeros and ones to any finite length string of zeros and ones, in such a way that it is computationally difficult to find an input that generates a desired output, or indeed tell $H(s)$ apart from a uniform random element of $\{0,1\}^\ell$.

Date: February 6, 2018.

To understand how such a function may be used, consider two co-authors who would like to choose a random order of authorship, and would like to prove that they indeed chose at random. One way to achieve that would be to instate a norm in which, for some fixed hash function H, the authorship order is given by the first bit of $H(s)$, where s is the concatenation of the authors' names and the title of the paper.

The issue with this idea is that the authors, who may be interested in a particular order, could try a number of different paper titles, until the resulting order suits their preferences. Our proposed mechanism makes it possible for the authors to publicly receive a random order, while making this type of manipulation impossible.

Another possible approach would be to have some central repository publicly record a request by the authors to choose a random order for a given paper, and then produce this random order for them, again recording it publicly. Creating such a public record would indeed alleviate concerns that the authors tried multiple paper titles. However, the problem of trust is now simply shifted to this central repository, who could be suspect of, for example, accepting pay to produce a desired outcome.

1.2. **Private and Public Randomness.** We distinguish between *public randomness* and *private randomness*. In public randomness, the randomness is requested by a user and delivered to her publicly. In private randomness, the user receives a public message, from which she can obtain a random input privately.

As an example, consider a auditor who wants to commit to choosing at random whom to audit. This may be important in settings of airport screenings, tax audits etc., in which the auditor wishes to avoid being accused of discrimination or other ulterior motives.

Obviously, the auditor does not want it to be known in advance who the targets of the audits will be. Our mechanism allows for the auditor to publicly commit to using a random input on a certain date, for the randomness to be delivered before that date, and for the auditor to be able to prove, after performing the audit, that it indeed used this randomness.

1.3. **Our solution.** Our mechanism makes use of a *public ledger* or *blockchain*. This is a well established decentralized mechanism that allows parties to leave a permanent public record in a searchable database, which is accessible to anyone with an internet connection; we explain more technical details below.

In our mechanism, a user submits, through a central "Randomness Authority", or briefly `RA`, a request for randomness, together with a

description of what it will be used for. This is publicly recorded by the RA in the blockchain. The RA then returns to the user a random input rand, which also depends on subsequent records (or *blocks*) in the blockchain. This returned input rand is public: it can be calculated by anyone with access to the blockchain. The dependence of rand on subsequent chains makes it impossible for the user to predict the input in advance. If private randomness is needed then the user uses a private key to calculate another random input from rand. She can choose to later reveal the key—which was committed by her before rand was known—making her random input public. A mechanism is in place to ensure that she indeed revealed the private key that she used.

In the co-author example, the authors, through the RA, would record, for example, their names and the title of their paper in the blockchain, leaving a permanent public record. The RA would then return them a random bit rand, which they would use to set the authorship order. The public record would allow anyone to verify that they indeed chose the order prescribed by the RA, and that they did not attempt this multiple times, with different title variants.

In the audit example, the auditor would choose a hard-to-guess private key s for each audit date. It would send the RA the hash $H(s)$, which the RA would record publicly, together with the date in which the generated randomness is to be used. The RA would then send the auditor back a random input rand, which the auditor would use, together with s, to determine the target randomness rand$'$. After the audit, the auditor would publish s, making it possible for anyone to verify that indeed the input given by the RA was used. Note that due to the properties of the hash function H it is computationally hard (and in practice impossible) to calculate s from $H(s)$, and so the targets would not be revealed when the request for randomness was made public. Also, it is hard to find an s' such that $H(s') = H(s)$, and so the user cannot manipulate s after revealing $H(s)$.

An important feature of our mechanism is that the RA need not be trusted: it is impossible for the RA to collude with the user, collude against the user, or more generally manipulate rand in any way.

2. Protocol

2.1. **Overview.** Our protocol is based on the power of a blockchain (a public ledger). Informally, a blockchain is an ordered chain of blocks, and a block is a list of records (transactions). Every fixed time interval t a new valid block is added to the end of the blockchain. Records can be added by any user (perhaps at a cost) and are publicly observable.

One important property of a blockchain is that a record (transaction) cannot be modified or deleted from a block that was added to the blockchain, and blocks that have been added to the blockchain cannot be modified, deleted or even moved. This property of blockchains allows users to make commitments that are publicly verified. A second important property of the blockchain is that the value of the next block cannot be manipulated by any one user or even a (not too) large group of users. These properties have made it possible to implement virtual currencies such as bitcoin using blockchains.

In our mechanism, the central Randomness Authority `RA` will maintain a database of users and their requests. It will also record these requests on a blockchain, and use the blockchain to calculate the random inputs requested by the users.

2.2. **Implementation.** The protocol is parametrized by a number $\ell \in \mathbb{N}$ and a hash function $H\colon \{0,1\}^* \to \{0,1\}^\ell$. Conceptually, it is convenient to think of H as an *ideal hash*, or a function that for each $s \in \{0,1\}^*$ returns a uniformly random (and then fixed) $H(s) \in \{0,1\}^\ell$. In practice, we would make used of some fixed, publicly known function, such as SHA-3.[1] We assume that H of the empty string equals 0^ℓ.

(1) User u sends a randomness request to the `RA`. The request contains three fields:

$$\mathsf{request} = [\mathsf{ID}, \mathsf{usage}, H(s)]$$

where

- **ID:** Identity of the user. I.e., a unique identifier connected to the user's public identity.
- **usage:** A commitment of usage. Namely, description of how, when and where the randomness will be used.
- **s:** Private randomness source. $s \in \{0,1\}^\ell$ is a string that will be used in generating private randomness. s will be the empty string when the request is for public randomness.

(2) The `RA` publicly posts the request on its website or other media.

(3) The `RA` submits the following transaction to the blockchain:

$$\mathsf{Trx} := [\mathsf{ID}; H(\mathsf{usage}); H(s)]$$

where ID, usage and $H(s)$ are from the request.

(4) Let B be the verified block in which Trx appears, and let B' be the following verified block. The `RA` sends back to the user the

[1] See https://en.wikipedia.org/wiki/SHA-3.

publicly verified randomness[2]

$$\mathsf{rand} := H(\mathsf{Trx}||H(B')).$$

Note that rand is publicly known.

(5) The random input to be used by the user is[3]

$$\mathsf{rand}' = \mathsf{rand} \oplus H(s||s).$$

Note that this is not publicly known, unless s is publicly known. This is the case when s is empty, in which case $H(s||s)$ is equal to 0^ℓ, and so $\mathsf{rand}' = \mathsf{rand}$.

When s is non-empty, and when the user want to prove that she used rand', she publishes s (possibly on the blockchain). This makes it possible for anyone to calculate rand'.

3. Applications

3.1. **Co-author ordering.** In [5], the case is made for choosing a random order for co-authors on academic papers, coupled with a symbol that appears in the paper and indicates that the order was selected at random. This is motivated by a number of studies that empirically find significant benefits for being listed earlier in the list of authors [1, 2, 8].

Of course, co-authors may have strategic reasons to declare that the choice was made at random when it was not. For example, when a senior advisor co-authors a paper with a student, both may prefer to have the student be the first named author, to help the student's career prospects. Here verified randomness can alleviate such concerns. The author $\mathtt{A}$ working on a paper P with co-author $\mathtt{Z}$ would send the request

$$\mathsf{request} = [\mathtt{A},\ \mathtt{A}||\mathtt{Z}||P||C,\ 0^\ell],$$

where C is the computer code that will be used to calculate the order of authorship from rand'. Since in this case s is known, $\mathsf{rand}' = \mathsf{rand}$ is public, and thus it is publicly verifiable that the authors used the order given by $C(\mathsf{rand})$.

3.2. **Lotteries and raffles.** State run lotteries have traditionally used public physical randomization devices, such as ball machines, to prove that their draws are random. These have been successfully manipulated in the past [7]. In the United States some state lotteries pick numbers using computers, and these too have indeed been vulnerable to manipulation by criminals (e.g., [4]).

[2]Here and below $\cdot||\cdot$ denotes string concatenation.

[3]Here $\oplus$ denotes the exclusive or operation.

Many companies run raffles as marketing promotions, and these are prone to the same weakness. Indeed, from 1995 to 2001, an employee of the company that ran McDonald's "Monopoly Game" promotion manipulated the results, stealing more than $13M in the process [6].

The use of verifiable public randomness in lotteries and raffles is straightforward and would alleviate such problems. In this application, the lottery L would, for each draw scheduled for date d, send to the RA the request

$$\mathsf{request} = [\mathtt{L}, d||\mathtt{C}, 0],$$

where C is the computer code that will be used to calculate the winning numbers on date d using the random input rand that will be generated from this request. This request would be submitted on the day of the draw, and once it is publicly recorded[4] it would be possible for anyone to calculate the winning numbers, and verify that those published by the lottery are indeed equal to $C(\mathsf{rand})$.

3.3. **Audits and inspections.** Random audits are a standard practice in security, tax, accounting, financial, electoral and other settings. It is often important for the auditor to prove that audits targets are chosen at random, in order to avoid accusations of discrimination, political persecution, corruption etc.

For an auditor A to use our mechanism in order to determine the targets of an audit scheduled for date d, the auditor would choose a random s and send the request

$$\mathsf{request} = [\mathtt{A}, d||\mathtt{C}, H(s)],$$

where again C is the code to be used in determining the target from rand'. After inspecting $C(\mathsf{rand}')$, the auditor can publish s, making rand' public, and making it possible for anyone to verify that the target was indeed $C(\mathsf{rand}')$.

Many disarmament treaties in history involved random inspections, to ensure compliance with the treaty terms. When the inspecting party has intelligence regarding the inspected party's activities, it may wish to deviate and not choose at random where to inspect. In such a setting, to ensure that inspections are random while not revealing them in advance, the inspecting party can use the same mechanism to generate private randomness, which can later be proved to have not been manipulated.

[4] In fact, one would need to wait until the *next* block B' is recorded.

3.4. **Medical experiments.** In many medical (and other) experiments, a group of subjects is randomly divided into a treatment group and a control group. The experimenter, who may have prior knowledge about the subjects, may have an incentive to choose these groups in a way that could influence the experiment outcome. For example, when testing the efficacy of a drug in treating a disease, the experimenter could choose the healthier subjects to be in the treatment group, with the obvious end result being that this group is healthier than the control group, despite the drug having no effect. Indeed, an existing practice is to choose a random sample again and again until a favorable one is chosen [3].

Using our mechanism, the experimenter would be able to prove to customers and regulators that no such manipulation has taken place, and the treatment and control groups were chosen at random.

When sensitive business and medical information is involved, the user may wish that not only the randomness rand' be (initially) kept private, but also the usage. In this case the user $\mathtt{U}$ can submit a request

$$\mathsf{request} = [\mathtt{U}, D_1||H(D_2), H(s)],$$

where D_1 includes the non-sensitive description of the experiment, D_2 includes sensitive information, and $D_1||D_2$ is a complete description of how rand' is to be used. After the fact, D_2 can be revealed (perhaps selectively, to the regulating authorities) if the user wishes, making it possible to verify that rand' was indeed used as the user committed to using it.

3.5. **Surveys.** It is conceivable that in an election, a political party may wish to distort survey results in order to manipulate public opinion. One way to achieve this, which may be difficult to detect, would be to bias the sample by choosing people who are known to be more likely to give a desired answer. A similar motivation may exist in other settings where a survey is used to assess population preference or trait distributions. For example, a leader who would like to portray her country as richer than it is, could commission a survey of household wealth in which affluent geographical areas are over-sampled. The usage of our mechanism in this case would be similar to the ones described above, and would alleviate such concerns.

References

[1] Ruth Chambers, Elizabeth Boath, and Steph Chambers, *The A to Z of authorship: analysis of influence of initial letter of surname on order of authorship*, BMJ: British Medical Journal **323** (2001), no. 7327, 1460.

[2] Liran Einav and Leeat Yariv, *What's in a surname? the effects of surname initials on academic success*, The Journal of Economic Perspectives **20** (2006), no. 1, 175–187.
[3] Ben Goldacre, *Bad science: Quacks, hacks, and big pharma flacks*, McClelland & Stewart, 2010.
[4] Associated Press, *Man who wrote code for state lotteries pleads guilty in rigging scheme*, Washington Post (2017June). https://www.washingtonpost.com/business/economy/man-who-wrote-code-for-state-lotteries-pleads-guilty-in-rigging-scheme/2017/06/29/f123fb02-5cd9-11e7-a9f6-7c3296387341_story.html?utm_term=.202a8f623117.
[5] Debraj Ray and Arthur Robson, *Certified random: A new order for coauthorship*, National Bureau of Economic Research, 2016.
[6] David Stout, *Eight charged with rigging McDonald's prize contests*, New York Times (2001August). http://www.nytimes.com/2001/08/22/us/eight-charged-with-rigging-mcdonald-s-prize-contests.html.
[7] UPI, *4 enter not guilty pleas in lottery rigging case*, New York Times (1981February). http://www.nytimes.com/1981/02/12/us/around-the-nation-4-enter-not-guilty-pleas-in-lottery-rigging-case.html.
[8] C Mirjam Van Praag and Bernard van Praag, *The benefits of being economics professor A (rather than Z)*, Economica **75** (2008), no. 300, 782–796.