

Master 2 AI-Health

Mini-project

October 2026

A digital random drawing system can avoid any risk of manipulation or cheating—provided it is well designed. A key requirement is that anyone should be able to verify after the fact that the draw is indeed the result of chance. We describe below a system that meets these conditions. We assume the availability of a good public hash function h , which associates any digital file F with a fingerprint $h(F)$ —for example, a sequence of 256 bits. Such a function satisfies the three properties below:

- (a) h is deterministic, but its output shows no identifiable regularity; it can therefore be regarded as a reproducible random draw.
- (b) Modifying a single character in the file F submitted to h completely changes the resulting output.
- (c) The function h is not invertible: in practice, given a 256-bit sequence s , it is impossible to find a file F such that $h(F) = s$, and given a file G , it is impossible to find another file G' such that $h(G') = h(G)$.

We also assume the availability of a public function t , which maps any hash produced by h to a lottery ticket (grid). We further assume that t produces tickets that are “well distributed” over the set of all possible tickets: since the number of possible hashes is much larger than the number of possible tickets, we require that, when considering all outputs of t , each possible ticket is produced approximately the same number of times.

On the day before the draw (the last day on which tickets can be played) the lottery organizer publishes on a public webpage the value $R = h(D)$, the hash of a data item D , where D consists of 256 bits and it is freely chosen by the organizer. No one can recover D from R , since h is not invertible.

On the day of the draw, the organizer consults the first digital page J of a daily newspaper chosen in advance and computes its hash $S = h(J)$. He then computes the logical sum of D (known only to him) and S (known to everyone). This logical sum is defined as the sequence of bits obtained by performing an exclusive *OR* (*XOR*, in logic) between the bits of D and S , thereby mixing the two data sets. He then computes the hash $X = h(D \text{ XOR } S)$, and finally generates, using function t , the ticket $G = t(X)$. He then makes public the

value D and the ticket G , which is the winning ticket of the day. This procedure prevents any form of cheating.

- The organizer cannot cheat by knowing the ticket in advance or choosing it arbitrarily. Indeed, when D was chosen, the newspaper's first page J was not yet known. Consequently, he could not manipulate the choice of $G = t(h(D \text{ XOR } h(J)))$, which depends on J . Everyone can verify that the D chosen the day before was not modified in the meantime, because computing the hash of D must yield R , which was made public the day before.
- The newspaper cannot cheat, because when it publishes its first page J on the day of the draw, the data D is still known only to the organizer; therefore, the newspaper cannot manipulate the value of $G = t(h(D \text{ XOR } h(J)))$, which depends on D .
- No one else can cheat, because bets are closed before the day of the draw, at a time when J is not fixed and D is known only to the organizer. Moreover, anyone can verify after the fact that there was no manipulation or cheating, because once D and G are published, anyone can check that $G = t(h(D \text{ XOR } h(J)))$.

Objective: to develop a computer program, in a programming language of your choice, that implements the protocol described above. Working in pairs is encouraged.

This protocol was developed and implemented by Szrek2Solutions Society.