Algorithmic pseudorandomness in quantum setups

Ariel Bendersky,[1, 2, 3] Gonzalo de la Torre,[1] Gabriel Senno,[2] Santiago Figueira,[2, 3] and Antonio Acín[1, 4]

[1] *ICFO-Institut de Ciencies Fotoniques, The Barcelona Institute of Science and Technology, 08860 Castelldefels (Barcelona), Spain*

[2] *Departamento de Computación, FCEN, Universidad de Buenos Aires, Buenos Aires, Argentina.*

[3] *CONICET, Argentina*

[4] *ICREA–Institucio Catalana de Recerca i Estudis Avançats, Lluis Companys 23, 08010 Barcelona, Spain*

Many experimental setups in quantum physics use pseudorandomness in places where the theory requires randomness. In this work we show that the use of pseudorandomness instead of proper randomness in quantum setups has potentially observable consequences. First, we present a new loophole for Bell-like experiments: if some of the parties choose their measurements pseudorandomly, then the computational resources of the local model have to be limited in order to have a proper observation of non-locality. Second, we show that no amount of pseudorandomness is enough to produce a mixed state by computably choosing pure states from some basis.

PACS numbers: 03.67.-a, 03.65.Ud

An ubiquitous scenario in experimental labs nowadays consists of classical computers monitoring quantum setups. This also includes commercial applications of quantum technologies, such as, e.g., quantum key distribution protocols, where computers control the preparation and measurements of quantum states.

A classical computer, however, is not an arbitrary device; it is as powerful as the standard formal model of Turing machines [1]. A natural question is whether this fact has any consequence in the physics that will be observed in experimental setups controlled by computers. The main purpose of this work is to study this general question in the context of randomness generation. In fact, it is a very well-known result that classical computers are unable to produce randomness. On the other hand, nowadays there are efficient algorithms [2, 3] that produce 'seemingly random' binary sequences, with excellent statistical properties. Our goal is to understand whether the use of algorithmic sources of pseudorandomness instead of ideal randomness has any observational consequences.

Our main result is to identify two situations in which the use of pseurandomness has observational consequences. First, we show that, when it comes to Bell-like experiments [4–6] to test non-locality, if the measurement independence between the two parties [7–10] is achieved via private pseudorandom number generators, it is possible to construct a local model that leads to an observed violation of Bell inequalities. Second, we show that pseudorandomness is not enough to produce a mixed state as a classical mixture of pure quantum states. In other words, we show that if in a setup used to produce a proper mixed state as a classical mixture of pure quantum states, we replace the random source by a pseudorandom one, situations that initially were not distinguishable turn out to be so. This has direct implications for experimental setups since mixed states are commonly prepared in this way [11, 12].

Bell computability loophole – Non-locality is one of the most intriguing features of quantum mechanics [4, 6]. The standard Bell scenario is described by n distant observers who can perform m possible measurements of r possible results or outputs. The measurements are arranged so that they define space-like separated events. For the sake of simplicity we focus our attention on the standard bipartite 2-input 2-output scenario [13], although our considerations apply to any Bell scenario.

It is convenient for what follows to rephrase the standard Bell scenario in cryptographic terms, as in [14–16]. In this approach, Alice and Bob get the devices from a non-trusted provider Eve. The standard local models correspond to classical preparations in which the devices generate the measurement results given the choice of measurements following a deterministic assignment depending on possibly correlated classical instructions, but independently of the input chosen by the other party. Bell inequalities are conditions satisfied by all these preparations, even when access is given to all the measurement choices and results produced in previous steps [14]. In turn, quantum correlations, obtained for example by measuring a maximally entangled two-qubit state with non-commuting measurements, can violate these inequalities. The violation of a Bell inequality witnesses the existence of non-local correlations. In turn, this can be used by Alice and Bob to certify the non-classical nature of their devices. This cryptographic approach to Bell tests makes it easy to understand the implications of our results for device-independent protocols based on non-locality. But our results also apply to the standard context in which Bell inequalities were introduced, namely on the discussion about the possible existence of a local model explaining quantum correlations [4, 6]. There, the local model can be seen as the eavesdropper that tries to reproduce the observed correlations, possibly by exploiting loopholes in the implementation.

A crucial condition to derive any Bell inequality is that the measurement choices by Alice and Bob are random, in the sense of not being predictable by Eve, or the local model, at the moment of preparing the devices. In fact, if this condition is not met and the eavesdropper knows the measurement choices in advance, she can prepare by classical means any form of correlations between Alice and Bob, in particular not bounded by any Bell inequality. In what follows, we consider the scenario in which pseudorandomness is used to choose the measurement settings in a Bell test. We show that if at least one of the parties chooses the measurements following an algorithm, this gives rise to a new loophole, which we call the *computability loophole*, even under the assumption that that the algorithm is independent of the boxes Eve prepares, and all she knows about the algorithm is an upper bound on its run-time. For this loophole to apply, the boxes prepared by Eve have to communicate the inputs used in the previous rounds [14–16], as shown in Fig. 1. The main intuition is that if Eve, or more precisely the devices she prepared, is able at some point to learn the algorithm generating the inputs, she could use this information to produce a fake Bell violation.

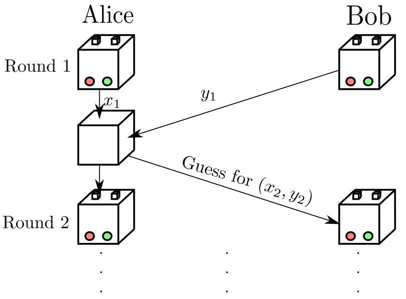


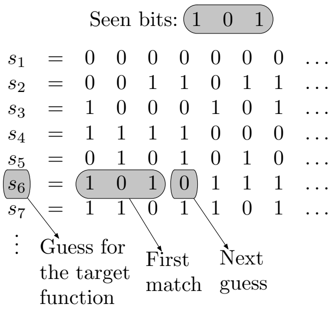
FIG. 1: Scheme for the Bell inequality computability loophole. After each round i, Alice's box receives Bob's last choice of measurement y_i. Using all previous choices of inputs for both parties, Alice's box makes a prediction for what the inputs of the next round will be by using the presented algorithm.

Let us assume that one party, say Alice without loss of generality, uses an algorithm to choose her inputs. In formal terms, this means that there is a computable function $f_A : \mathbb{N} \to \{0, 1\}$ such that $f_A(i)$ tells Alice to press the left (0) or the right (1) button at the i-th round.

It is clear that if Eve knows (any algorithm for) f_A, her task becomes trivial. Hence, we assume f_A is unknown to Eve when she prepared the boxes. Eve does not know either which party is using the function f_A to choose the inputs. However, we will assume the following further hypothesis: Eve knows *some* computable function t which upper bounds the running time needed to compute f_A. For instance, Eve knows that f_A is computable in, say, time $O(t(n))$, for $t(n) = 2^{2^n}$ —though the algorithm that Alice is actually running may take, say, $O(n^2)$.

Knowing this time bound t will allow Eve to program a computing device in one of the boxes, say Alice's, which will be able to *predict* f_A after finitely many rounds. This means that Alice's box will have an effective procedure that, after having seen $f_A(0), f_A(1), \ldots, f_A(k)$ for large enough k, will allow it to correctly guess $f_A(k+1), f_A(k+2) \ldots$. The existence of such k will be guaranteed by construction; however Alice's device will not be able to tell when this k has arrived.

The idea behind this is a standard result from computer science: algorithms whose running time is upper bounded by some computable function t can be enumerated in a computable fashion (see the Supplemental Material [17]). Using such an enumeration, Alice's device will pick at every round k the first algorithm whose output coincides with $f_A(0), \ldots, f_A(k-1)$ and use it to predict $f_A(k)$. Since, by assumption, the algorithm used by Alice will be eventually enumerated, choosing in this way, Eve is guaranteed to converge, after finitely many rounds, to an algorithm for f_A (maybe not the same as Alice's) and hence, start correctly guessing Alice's inputs. See Fig. 2 for a schematic description of the guessing protocol and the Supplemental Material [17] for the formalization.


Seen bits: 1 0 1

s_1 = 0 0 0 0 0 0 0 ...
s_2 = 0 0 1 1 0 1 1 ...
s_3 = 1 0 0 0 1 0 1 ...
s_4 = 1 1 1 1 0 0 0 ...
s_5 = 0 1 0 1 0 1 0 ...
s_6 = 1 0 1 0 1 1 1 ...
s_7 = 1 1 0 1 1 0 1 ...
⋮

FIG. 2: Suppose $\{s_i\}_{i\in\mathbb{N}}$ is a (computable) enumeration of the algorithms which run in $O(t)$ time. After seeing $f(0) = 1$, $f(1) = 0$ and $f(2) = 1$, the guess for $f(3)$ will be done with the first algorithm whose outputs match those values (in the example, s_6.)

At this point, we can further clarify the need to assume a bound on the complexity of f_A. As we saw, the loophole is based on the ability to program a predictor (in the sense described above) for functions belonging to a given class. It is a basic result in computability theory that the class of all computable functions is not predictable [18]. We could have chosen other ways to restrict the class of functions, but computational resources seemed to be rather natural.

Despite being necessary for the protocol, one can justify the time complexity assumption on the following grounds:

1. it is natural to require that the time Alice and Bob take to choose their measurements on each round is bounded, and

2. the number of computational steps per second that a physical system of mass m can perform is upper bounded by $2(mc^2)/\pi\hbar$ [19].

These two facts imply that the number of computational steps that Alice's and Bob's algorithms can take on each round n is bounded by a constant and hence, their computational complexity is, at most, linear in n (and, so, exponential in $|n|$, the size of n).

Eve is able to prepare both boxes so as to fake a Bell inequality violation. To see how, notice that any no-signaling bipartite probability distribution, local or not, can always be written as

$$\begin{aligned} P(a,b|x,y) &= \int p(\lambda)\delta^a_{f(x,\lambda)}\delta^b_{g(y,x,\lambda)}d\lambda \\ &= \int p'(\lambda)\delta^b_{f'(y,\lambda)}\delta^a_{g'(x,y,\lambda)}d\lambda \end{aligned}$$

where again functions f, f', g, g' are deterministic functions. This means that, given that Eve learns either Alice's input x or Bob's input y, she can prepare deterministic (local) boxes to simulate any probability distribution and hence fake any Bell violation.

Regarding the complexity of Eve's protocol, there are two measures that one can study. First, there is the number T of time steps that it takes Alice's box to make a guess: if Eve assumes an upper bound of $t(n)$ for the running time of Alice's algorithm, then $T = O(t(n)\cdot\log(t(n))$ (for $t(n)$ time constructible, see [20, §1.3]). Second, there is the number M of mistakes that Alice's box will make before starting to guess correctly. Using the *halving algorithm* of Barzdin and Freivalds (see [21, Thm. 6]), the learning process can be carried out in such a way that $M \leq O(\max(l, \log(c)))$, where l is the length of Alice's algorithm and c is such that it runs in time $c \cdot t(n)$.

This means that Eve will not require too many rounds, in terms of l and c, to fake non-locality. That is, if we look at the distribution generated in the first n rounds, the fraction of inputs-outputs that will not serve Eve's purpose of faking a non-local distribution is upper bounded by M/n, which vanishes with increasing n. So, if Alice wants to make this number of rounds large, then she either has to use a very long program or an enormous time constant.

It is relevant to place these considerations in the context of recent "loophole-free" Bell experiments [22–24]. In all these experiments the choice of measurements was performed using the fast quantum random-number generator of [25]. Thus, assuming the validity of quantum physics, these experiments are free from the computability loophole introduced here. However, one may argue that it is rather undesirable, and even circular, to depend on the validity of a non-local theory, such as quantum physics, to test non-locality. The use of random numbers of quantum origin is better justified in device-independent protocols based on non-locality, as the validity of quantum physics is assumed for many of them.

Pseudorandom mixtures – Our second result concerns preparations of mixtures of quantum states using pseudorandomness. Let us consider the following game: Alice has access to a computer in an unknown configuration and with unbounded memory that is running a presumably very convoluted unknown algorithm to generate a pseudorandom sequence of bits. Alice encodes the generated bits into quantum states, either on the single qubit eigenstates of σ_z, $\{|0\rangle, |1\rangle\}$, or the eigenstates of σ_x, $|\pm\rangle = \frac{1}{\sqrt{2}}(|0\rangle \pm |1\rangle)$. The choice of basis is made by Alice at the beginning of the protocol and kept fixed. The resulting qubit states are sent to Bob as seen in Fig. 3. Bob's goal is to guess the basis used by Alice to encode the sequence.

Alice's algorithm is completely unknown. Hence, after having seen only finitely many qubits, Bob has no a priori reason to favor any of the two alternatives for the next qubit. Bob could be tempted to assign uniform probability to these events and thus characterize the situation with the maximally mixed state $\rho = \frac{I}{2}$, consequently giving undistinguishable situations for the σ_z and σ_x preparations. This is in fact what the theory predicts if the preparation is properly random.

In the following, however, we show that the fact that the preparation procedure was performed in a computable way leaves a trace which allows to distinguish both situations in finite time and with arbitrarily high success probability. This implies that characterizing Bob's lack of knowledge about the state of the systems coming out of the box with $\rho = \frac{I}{2}$ is incorrect. It is worth mentioning that having a computer mixing the state does not imply that the sequence used to prepare the state is periodic. In fact, there exist computable sequences which not only avoid periodicity but also have good randomness properties; we can mention, for instance, Borel *normal* sequences or *polynomial-time random* sequences [3].

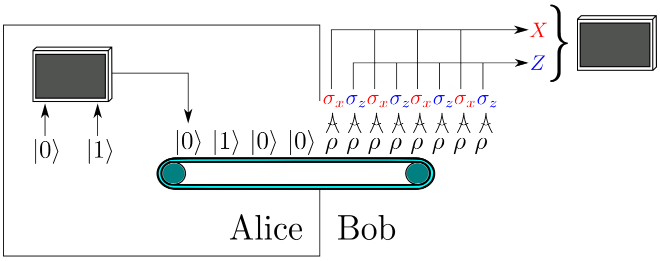

X
Z
$\sigma_x\sigma_z\sigma_x\sigma_z\sigma_x\sigma_z\sigma_x\sigma_z$
$\rho\ \rho\ \rho\ \rho\ \rho\ \rho\ \rho\ \rho$
$|0\rangle$ $|1\rangle$
$|0\rangle$ $|1\rangle$ $|0\rangle$ $|0\rangle$
Alice
Bob

FIG. 3: Alice uses a computer to choose between $|0\rangle$ and $|1\rangle$ (or $|+\rangle$ and $|-\rangle$), keeping the basis fixed all through the experiment. To distinguish both possible preparations, Bob measures alternatively σ_x and σ_z and feeds the resulting sequences to a computer executing Algorithm 1.

The measurement strategy allowing Bob to make a correct guess works as follows. Bob measures every qubit that comes out of the black box on an odd position in the basis of eigenstates of σ_z and every qubit that comes out on an even position in the basis of eigenstates of σ_x yielding two binary sequences of measurement results Z and X respectively, as can be seen in Fig. 3. The one corresponding to the choice of measurement that matches the preparation basis is computable, and the other one corresponds to a fair coin tossing, according to quantum mechanics. Therefore we need an algorithm that given two sequences, one being computable and one arising from a fair coin tossing, is able to tell us which is which in finite time and with an arbitrarily high probability of success. Contrary to the previous situation, our result is to show that this algorithm exists.

To distinguish which of the two sequences is computable we borrow some tools from the theory of algorithmic randomness [26–28]. Roughly, an infinite binary sequence is random in an algorithmic sense, if it lacks any regularity. Randomness tests, also called *Martin-Löf tests*, are defined to detect some specific regularity. Therefore a sequence is algorithmically random if it fails every possible Martin-Löf test. This 'detection' of non-random sequences should be computably approximable, with incrementing levels of accuracy or significance. At level m of significance, a given test V describes a set of possible prefixes of sequences that don't look random (namely, a set V_m). As we move on with the level of significance m, each test rules out more sequences leaving in the limit a null measure set of non-random sequences. The algorithmically random sequences are those ruled out by every possible Martin-Löf test.

Formally, a Martin-Löf test (ML test) is a sequence $(V_m)_{m\in\mathbb{N}}$ of sets of binary strings with two properties:

1. *Effectiveness.* There is a program that given m and i, produces the i-th string of V_m (notice that in general there are infinitely many strings in V_m). It is not possible to computably determine if a string *is not* in V_m, but we can computably enumerate all strings that are in.

2. *Null class.* If A is a set of binary strings, then $[A]$ is defined as the set of infinite sequences with prefixes in A. If λ is the uniform measure on the space of infinite binary sequences, each ML test $(V_m)_{m\in\mathbb{N}}$ should satisfy $\lambda[V_m] \leq 2^{-m}$.

Let Y be an infinite sequence. Y is ML random if no test $(V_m)_{m\in\mathbb{N}}$ can capture Y in *all* its levels of accuracy, that is if for no test $(V_m)_{m\in\mathbb{N}}$ we have $Y \in \bigcap_m [V_m]$. Informally, if $Y \in [V_m]$ then we reject the hypothesis that Y is random with significance level 2^{-m}.

Let $Y \upharpoonright n$ denote the prefix of length n of the sequence Y. Observe that if $Y \in [\{\sigma_1, \sigma_2, \dots\}]$ then for large enough n we have that all the infinite sequences extending $Y \upharpoonright n$ belong to $[\{\sigma_1, \dots, \sigma_n\}]$. This last expresion can be seen as the n-th approximation of $[\{\sigma_1, \sigma_2, \dots\}]$. Hence if $Y \in \bigcap_m [V_m]$, then for every m there is n such that any extension of $Y \upharpoonright n$ is included in the n-th approximation of $[V_m]$.

It is well known that there is an universal ML test $(U_m)_{m\in\mathbb{N}}$ such that Y is ML random iff $Y \notin \bigcap_m [U_m]$. Since $\lambda \bigcap_m [U_m] = 0$, the set of ML random sequences has measure 1. In other words, the output of a fair coin tossing forms a ML random sequence with probability 1. On the other hand, the universal test $(U_m)_{m\in\mathbb{N}}$ detects any effective pattern. In particular any computable sequence fails it.

We return to the protocol that Bob follows to distinguish the computable sequence from the random (coin tossing) one: given a significance level 2^{-m}, he starts enumerating all the stings in $U_m = \{\sigma_1, \sigma_2, \dots\}$ until he finds some n such that for $Y = X$ or $Y = Z$ we have that all extensions of $Y \upharpoonright n$ belong to $[\{\sigma_1, \dots, \sigma_n\}]$.

Since either X or Z is computable, the last condition has to be satisfied for sufficiently large n. If the above condition was first satisfied by $Y = X$, he claims that X is the computable sequence and that Y is the random one; if the above condition was first satisfied by $Y = Z$ he claims he claims that Z is the computable sequence and that X is the random one. This decision is wrong when the random sequence was captured by $[V_m]$ before the computable one was (of course, for some $m' > m$ the random sequence would be out of $[V_{m'}]$). Hence, the probability of making this error is at most the probability for the coin tossing sequence to be inside $[V_m]$, and this is at most 2^{-m}.

Observe that in the above protocol there is nothing special with one of the sequences being computable. All that matters is that one of the sequences is not ML random. To study the effect of noise on the previous algorithm we considered a simple noise model described by a flip probability 1/2 in the observed symbols. This means that Bob does not receive the computable sequence Y, but a sequence Yxor N, where the xor is taken bitwise and N, the noise sequence, is an infinite sequence such that the limit relative frequency of the symbol 0 is strictly greater than the expected value, i.e.

$$\limsup_n \frac{\#\{i \mid N(i) = 1\}}{n} < 1/2.$$

This means an error ratio of strictly less than 1/2. It can be shown that N is not ML random, and that if Y is computable then Yxor N is also not ML random. Now, Bob can apply the same protocol as above to distinguish Yxor N, which is not ML random, from the one coming from the coin tossing.

The existence of the described distinguishing protocol proves that, if one uses a pseudorandom source instead of a random one to produce a classical mixture of pure states described by ρ, the resulting situation cannot be described by the same state ρ. On the down side, and although the specifics depend on the particular implementation of the universal test, the procedure is far from being efficient. However, for practical demonstrations, it is not difficult to see that by restricting Alice's class of algorithms, one can design efficient specific tests for Bob in the same spirit as the given above.

Discussion – In this work we presented two consequences of replacing randomness with pseudorandomness in quantum theory. First, we showed that if either Alice or Bob choose the inputs for a Bell experiment in a computable way, an eavesdropper preparing deterministic devices can make them believe they have non-local boxes, thus creating a loophole. For the loophole to apply, the boxes should communicate between rounds and adapt accordingly, as for instance studied in the context of the memory loophole [15, 16]. There is no way of preventing this form of communication, unless some assumptions regarding the shielding of the devices are enforced, or by imposing that all the measurements in the Bell test by one of the parties are space-like separated from those by the other party.

Second, we showed that if Alice uses a computer to prepare a seemingly proper mixture of $|0\rangle$ and $|1\rangle$ or a seemingly proper mixture of $|+\rangle$ and $|-\rangle$, Bob can distinguish both situations with arbitrarily high probability and without any access to Alice's algorithm. Our algorithm, although impractical, fulfils its purpose of showing that both preparations are indeed distinguishable. Our results imply that it is incorrect to model Bob's lack of knowledge in this scenario with independent copies of the maximally mixed state. They apply to, for instance, the mixed states experimentally produced using a classical random number generator [11, 12].

Acknowledgements – The authors gratefully acknowledge the comments and suggestions of anonymous reviewers that highly improved the presentation and results of this paper. This work is supported by the ERC CoG QITBOX, the AXA Chair in Quantum Information Science, the John Templeton Foundation, the Argentinian UBA (UBACyT 20020110100025), ANPCyT (PICT-2011-0365), the Laboratoire International Associée INFINIS, the Generalitat de Catalunya (SGR 875) and Spanish MINECO (Severo Ochoa Grant SEV-2015-0522 and FOQUS FIS2013-46768-P). GdlT acknowledges support from Spanish FPI grant (FIS2010-14830).

[1] P. Odifreddi. *Classical recursion theory*, volume 2 of *Studies in logic and the foundations of mathematics*. Elsevier, 1999.

[2] M. Matsumoto and T. Nishimura. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation (TOMACS)*, 8:3–30, 1998

[3] S. Figueira and A. Nies. Feasible analysis, randomness and base invariance. *Theory of Computing Systems*, 56(3): 439-464 ,2015.

[4] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 47:777–780, 1935.

[5] N. Bohr. Can quantum-mechanical description of physical reality be considered complete? *Phys. Rev.*, 48:696–702, 1935.

[6] J. Bell. On the Einstein Podolsky Rosen paradox. *Physics*, 1:195–200, 1964.

[7] J. Conway and Simon Kochen. The free will theorem. *Foundations of Physics*, 36(10):1441-1473, 2006.

[8] D. E. Koh, M. J. W. Hall, Setiawan, J. E. Pope, C. Marletto, A. Kay, V. Scarani, and A. Ekert. The effects of reduced "free will" on Bell-based randomness expansion. *Phys. Rev. Lett.* 109:160404, Oct 2012.

[9] Michael J. W. Hall. Local deterministic model of singlet state correlations based on relaxing measurement independence. *Phys. Rev. Lett.*, 105(25):250404–, Dec 2010.

[10] J. Barrett and N. Gisin. How much measurement independence is needed in order to demonstrate nonlocality? *Phys. Rev. Lett.*, 106:100406, March 2011.

[11] E. Amselem and M. Bourennane. Experimental four-qubit bound entanglement. *Nature Physics*, 5(10):748–752, 2009.

[12] J. Lavoie, R. Kaltenbaek, M. Piani, and K. J. Resch. Experimental bound entanglement in a four-photon state. *Phys. Rev. Lett.*, 105:130501, Sep 2010.

[13] J. F. Clauser, M. A. Horne, A. Shimony, and R. A. Holt. Proposed experiment to test local hidden-variable theories. *Phys. Rev. Lett.*, 23(15):880–884, Oct 1969.

[14] J. Barrett, D. Collins, L. Hardy, A. Kent, and S. Popescu. Quantum nonlocality, Bell inequalities, and the memory loophole. *Phys. Rev. A*, 66:042111, Oct 2002.

[15] S. Pironio, A. Acín, S. Massar, A. B. de La Giroday, D. N Matsukevich, P. Maunz, S. Olmschenk, D. Hayes, L. Luo, T. A. Manning, et al. Random numbers certified by Bell's theorem. *Nature*, 464(7291):1021–1024, 2010.

[16] S. Pironio and S. Massar. Security of practical private randomness generation. *Phys. Rev. A*, 87:012336, Jan 2013.

[17] See Supplemental Material at [URL will be inserted by publisher] for the technical details of the protocols.

[18] E Mark Gold. Language identification in the limit. *Information and control*, 10(5):447–474, 1967.

[19] S. Lloyd. Ultimate physical limits to computation *Nature*, 406(6799):1047–1054, 2000.

[20] S. A. Boaz Barak. Computational complexity: a modern approach. *Cambridge University Press*, 2009.

[21] T. Zeugmann and S. Zilles. Learning recursive functions: A survey. *Theoretical Computer Science*, 397(3):4–56, 2008.

[22] B. Hensen *et al.*. Loophole-free Bell inequality violation using electron spins separated by 1.3 kilometres. *Nature*, 526:682–686, 2015.

[23] M. Giustina *et al.*. Significant-Loophole-Free Test of Bell?s Theorem with Entangled Photons. *Phys. Rev. Lett.*, 115:250401 (2016).

[24] L. K. Shalm *et al.*. Strong Loophole-Free Test of Local Realism. *Phys. Rev. Lett.*, 115:250402 (2016).

[25] C. Abellán *et al.*. Generation of Fresh and Pure Random Numbers for Loophole-Free Bell Tests. *Phys. Rev. Lett.*, 115:250403 (2016).

[26] A. Nies. *Computability and Randomness*, volume 51 of *Oxford Logic Guides*. Oxford University Press, 2009.

[27] R. Downey and D. Hirschfeldt. *Algorithmic Randomness and Complexity*. Theory and Applications of Computability Series. Springer, 2010.

[28] M. Li and P. Vitanyi. *An Introduction to Kolmogorov*

Complexity and Its Applications. Springer Verlag, 2008.